

SINCLAIR

SEPTEMBER 1991 £1.95



4 6
7-10
11 13
14 16
19
25
29
34
40
44
46

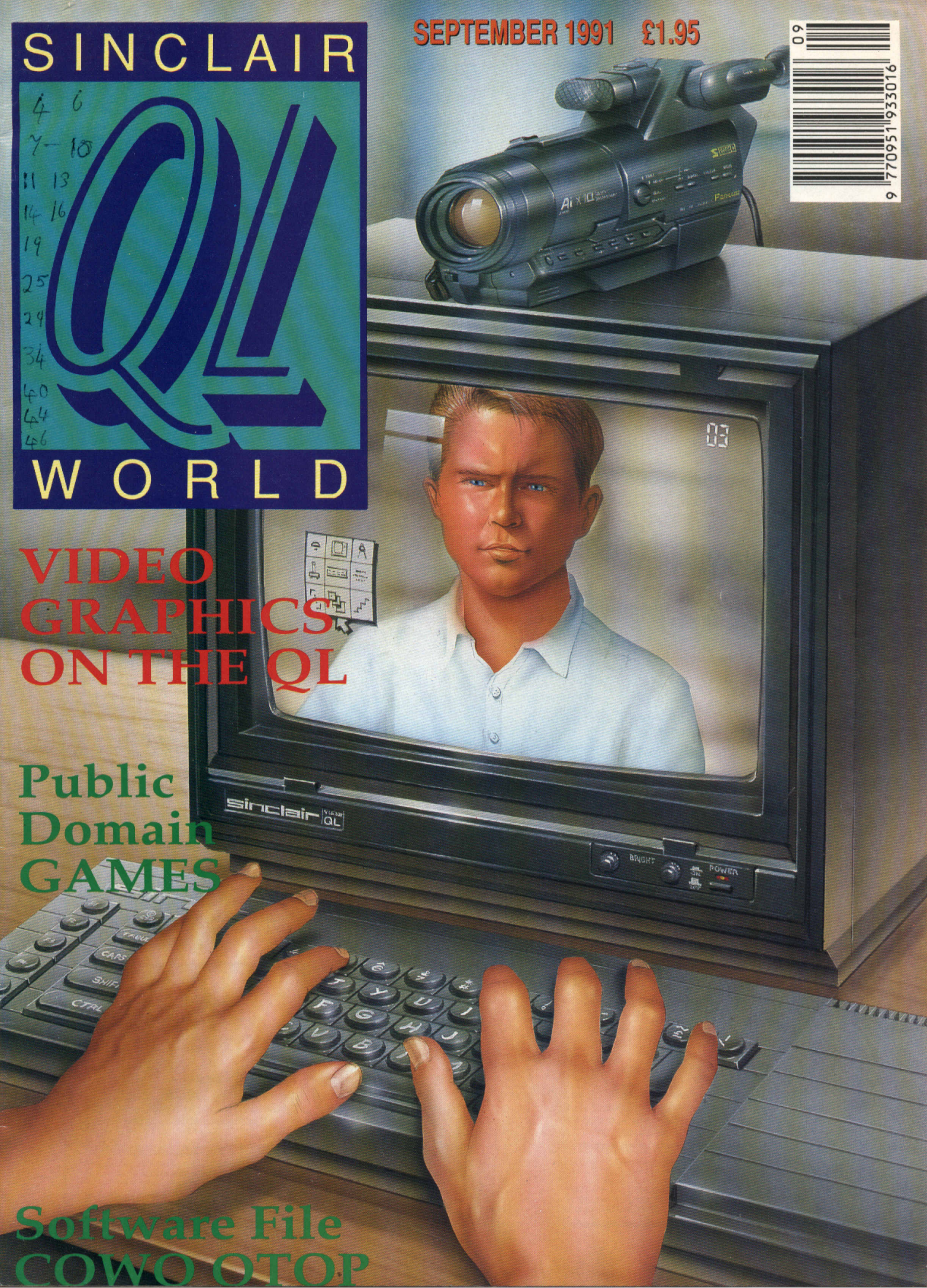
WORLD



VIDEO
GRAPHICS
ON THE QL

Public
Domain
GAMES

Software File
COWO OTOP



SINCLAIR



WORLD

Editor

Helen Armstrong

Production Controller

Jayne Penfold

Designer

Jeff Gurney

Advertising Manager

Jason Newman

Magazine Services

Yvonne Taylor

Advertising Production

Michelle Evans

Group Advertising Manager

Jean Dorza

Group Advertising Sales

Manager

Lynda Elliott

Group Editor

John Taylor

Deputy Managing Director

Ray Lewis

Managing Director

Peter Welham

Sinclair QL World
Panini House
116-120 Goswell Road
London EC1V 7QD
Telephone 071-490 7161
ISSN 026806X

Unfortunately, we are no longer able to answer enquiries made by telephone. If you have any comments or difficulties, please write the The Editor, Open Channel, Trouble Shooter, or Psion Solutions. We will do our best to deal with your problem in the magazine, though we cannot guarantee individual replies.

Back issues are available from the publisher price £2 U.K., £2.75 Europe. Overseas rates on request.

Published by Maxwell Consumer Magazines, A Division of MCPC Ltd., Sinclair QL World is distributed by IPC Market force, King's Reach Tower, Stamford Street, London SE1 9LS. Subscription information from: MSM Subscription Dept, Lazahold Ltd., PO Box 10, Roper St, Pallion Ind. Est., Sunderland SR4 4SN. Tel: 091 510 2290 UK: £21.00, Europe: £32.00, Rest of the World: £38.00.

Typesetting by Ford Graphics, 8-10 Whitsbury Road, Fordingbridge, Hampshire. SP6 1BR. Tel: (0425) 655657. Printed and bound by BPCC, Colchester. Covers printed by Spottiswoode Ballantyne, Colchester. Sinclair QL World is published on the third Thursday preceding cover date.

© COPYRIGHT
SINCLAIR QL WORLD - 1991

CONTENTS

SEPTEMBER 1991

- 4 TROUBLESHOOTER ● Perfection and Quill**
- 13 QL SCENE ● Qimi mouse back in action**
- 14 OPEN CHANNEL ● Plugs wanted**
- 19 QL SCENE ● Italy clubs link up**
- 20 VIDEO GRAPHICS ON THE QL ● Building up a system**
- 25 THE NEW USER GUIDE ● Part 7 - Subsystems**
- 29 SOFTWARE FILE ● Cowo Qtop**
- 32 PUBLIC DOMAIN GAMES ● Low cost games in the UK**
- 34 DIY TOOLKIT ● More of MORE**
- 38 QUESTIONNAIRE ● Come in, the other 98%**
- 40 ARCHED ● Part one of a simple SuperBasic database**
- 44 SOFTWARE FILE ● Winback**
- 47 CLUB ACCESS**
- 48 MICRODRIVE EXCHANGE**
- 49 NOTICE BOARD**
- 50 INSTANT ACCESS**
- BACK COVER - SUBSCRIPTION INFORMATION**



NEXT MONTH

ONE PER DESK

A look at the current position of a close QL cousin

DBQL PART 4

Continuing our relational database project

T A P R O U B L E

One of this magazine's forebears, was called *QL User*. What's in a name? Obviously, a second magazine for QL folk had to have a different name, and the first real rival was named *QL World*. The choice of 'World' gives a wider sound to the title, and that is the name which stuck when elements of both magazines were merged to form the current one. The QL is a computer which has found supporters all around the world (as did its Sinclair predecessors – ZX80, ZX81, Spectrum etc – and successor the Z88). Despite the QL having been out of production for several years, there is still a lot of enthusiasm.

We are all QL users, but the perception is that QL owners tend to be divided into two main categories – 'users' and 'hackers'. For the latter, the computer itself is perhaps the main interest, whereas people in the former category can be expected to be more concerned with the results they get out of it. These are clearly rather nebulous definitions. Someone who writes programs for their own sake, with no intention of using them for particular purposes subsequently, is nevertheless interested in results; it is just a question of emphasis. Some people are more fired-up by the mechanics of hardware and software, others want only to use them to do specific jobs.

Look and feel

For those who are users in the sense of not being too concerned with the technical aspects of how information is put on the screen, or out to a printer, the 'look and feel' of software is very likely much more important than the hardware, as long as the hardware does a reasonable job. It is only if the hardware has significant failings that this type of user takes a serious interest in it. An obvious case is the tendency for QL to lock in a hot environment or where there is electrical interference through the mains power supply. If you don't have such problems, you are more bothered by bad behaviour of software.

What prompted this line of thought was a day spent working solely on a pc, during which time it locked up several times. It suddenly struck me that this was behaviour which would have driven me mad in the early days of QL ownership, and caused me to spend many, many hours investigating ways of making the hardware more reliable, and writing to software suppliers to complain about program problems.

Bryan Davis looks further at Perfection, the successor to Quill in the QL market, and the recent All Formats fair in London.

My QL, and the software used on it (by and large), have long since become stable, with lock-ups being rare and crashes almost a thing of the past. Maintenance, in the widest sense, takes up an insignificant proportion of my QL time, whereas it seems to be taking of the order of 10-20% of my pc time. The lock-ups on the day in question were probably caused by program bugs; the database program in question has many bright features, but was never fully sorted out. In the preceding few days, I'd had a spate of hard disk errors, and more than two years of checking have not resulted in my pinning down whether this is a hardware, or a software, problem.

At the time the last few sentences were written, there were no evident problems with my QL system, but the hard disk has been reconnected since then and now all the files on it are 'not found'. Here we go again. . . reformat the disk, and recopy the dozens of files that were on it.

Easy to use

The program you use most may not be the 'best' one. You may simply have got used to it, and not be sufficiently bothered by its weaknesses to replace it by another. *Quill* is an excellent example of a program which survives almost in spite of itself. It causes many users all sorts of agony, but it remains the choice for the majority. After all these years this simply cannot be because it was supplied free. Above all things, it is easy to get to know, and easy to use afterwards. It doesn't require hours of study on the part of the user, to be able to do simple jobs. While thinking this, there lurks in the back of my mind the amount of effort that Digital Precision has been putting into its Quill competitor, *Perfection*. As with Software⁸⁷ and *text*⁸⁷ there are many reasons to ar-

gue that the concept and the program are far better than Quill, but what sells the product is as much the user interface as the functions provided, and it is this area that DP and Software⁸⁷ have been devoting attention to recently.

Key combinations

There are roughly one hundred functions in *Perfection* for which DP has had to find suitable key combinations. Given the available combinations, and the fact that many users already allocate the ALT key plus alphanumerics to certain functions (eg ALT-D to print the date on to the screen), there was a considerable problem to be solved. The desirable situation, of using keys with mnemonic connection to the functions (such as P for printing), has been obtained for main functions through the Quill-style main menus, and this should meet with approval from users.

Quill users should be able to make reasonable use of *Perfection* by referring to the menus only, without having to read the instructions to get going. Anyone not wanting to spend the time to learn how to use unfamiliar (to Quill users) functions can ignore them and make use of only those functions previously used in Quill, while revelling in the much better performance of *Perfection*. That is, it can be treated as no more than the fast Quill that was being asked for right from day one of the QL. You don't get major improvements for nothing, though, and you'll have to spend time with the instructions to be able to make full use of the program – and the instructions are DP's normal 'full-size' type!

To some extent, the user, as opposed to the hacker, isn't bothered what computer is being used. It's the program that matters. However, there are some fairly basic needs that a computer must fill to satisfy demanding users.

One of these is the ability to switch from one program to another without the first one being 'killed' in the process. It was enough for most of us to be able to use the Psion Quartet one at a time, back in 1984, but the restriction of one program at a time very soon got irksome. It wasn't the QL's fault; it was capable, right from its introduction, of having several programs loaded at the same time, and even of having some of them actually working concurrently. The obstacle was

SHOOTER

M S O L V E D

the way the Psion programs were written. They all required the EXEC_W command to start them, and that excluded the use of the CTRL-C keying to switch to any other executable program, or even to SuperBasic. On reflection, it is amazing that no one – so far as I know – published a way of making the Psion programs start with EXEC instead of EXEC_W almost as soon as the QL came out. Now, we have several ways of switching between programs and, essentially, all major QL programs can be run together during one session, subject to whatever memory limitation exists in the particular QL.

2MB document space

It is possible with some programs (eg Quill) to make use of disk space when memory runs out, so that one data file in one program growing too big may not necessitate calling a halt to the session. Perfection doesn't permit this, but allows documents to be as big as about 2MB, without it being necessary to configure the program to take a specific amount of memory beforehand; there aren't going to be many users who can't make do with this size. Referring back to the PC database again, it is one of those infuriating programs that won't make full use of the memory provided; it will say 'out of memory' when there are several MB of memory unused. Making use of a switching program makes matters worse with big data files, because this database program then won't even load them if they are above a certain, minimal size (the switching program takes vital memory for its own use).

On the QL, we now have the ability to run pretty well whatever combination of programs we fancy, and – with the Gold Card – to be generous in memory allocations to them, and to allow some programs to overflow their data files to disk if things get really tight. All this, plus real speed, with the Gold Card. Things have never looked better.

Bigger drives

Miracle Systems indicate that the 3.2MB high-density 3 $\frac{1}{2}$ in drives for the Gold Card may not be too far away. It is a question of Miracle obtaining them at the right price, and negotiations have a more hopeful air now. Orders for the *Gold Card* have been

sufficient to create a waiting list, with four to six weeks being quoted from the Miracle stand at the recent All Formats computer fair on 22 June.

Quanta helps

There was a goodly amount of activity around the QL suppliers' stands at this fair. Miracle were demonstrating the Gold Card, EEC had the PC keyboard and interface, TF Services had a neat little programmed robot, DJC had the two hard disk programs *WinBack* and *The Gopher*, Digital Precision had the usual audience for demonstrations of Perfection (much midnight oil having been burnt working on the instructions the previous nights), and the Quanta users' group seemed to be dealing with plenty of enquiries.

Thinking of Quanta, Chairman Phil Borman has produced some very handy additional commands for use with hard disk drives (eg Miracle and Rebel types), and these commands will be made available to Quanta members through their software library. The Miracle hard disk software is very sparse and leaves the user rather in the dark about what is going on. Phil has written a routine to give a map of the directory 'tree' on the hard disk, and another to allow a Path command (rather as in MS-DOS); other commands make changing sub-directories much simpler. In general, these utility functions appear to make use of the hard drive much more pleasant; I should be able to comment upon them in more detail shortly. It may be that Miracle have it in mind to make some similar improvements.

In case it has skipped your notice as well as mine, you now have to be even more careful when talking about disk drive sizes. What has been the standard size for 3 $\frac{1}{2}$ in drives over the past few years – the one-third-height form factor – has been superseded by one-quarter-height, and this is the size suppliers are now likely to supply, and talk about. One-third-height drives are about 3-4cm high (similar in height to a 50 pence coin in the UK), whereas one-quarter-height are about 2-3cm. Apart from the physical considerations, such as whether a new drive fit into an existing slot or casing, there will be electrical differences to consider. While the connectors and interface specifications may still be the same, the behaviour of the drive mechanism could be different. This became apparent when

a pre-production Gold Card took exception to my one-third-height Mitsubishi drives, but is apparently quite happy with one-quarter-height ones of the same make (and with one-third-height NEC drives).

It is sensible to specify your drive make and size (both disk diameter and drive height) when ordering an interface. The formatted capacity of the QL extra-high-density drives will be 3.2MB, whereas the PC version will be 2.88 (720KB x 4); the basic, unformatted capacity in each case is 4MB. Hopefully, the actual drive mechanism will be identical for the two computers. If you want to use the *Conqueror* MS-DOS emulator on the QL with the extra high density drives, you'll need to have a copy of MS-DOS 5.0, which should be available before the end of 1991. Digital Precision will be looking at making any necessary changes to the Conqueror package to permit use of these drives.

Have you noticed that the mechanism of hard disk drives has been reduced so much in size that a 40MB drive can now be had which is much smaller than a one-quarter-height 3 $\frac{1}{2}$ in floppy drive? The drive is 2in type, made for laptop computers, where space is at a premium. In fact, a 40MB 2in drive that has a smaller area than your typical plastic charge card is available.

Quill uninterested

Thinking of floppy drives, it was so long since I had used Quill seriously that my recollection of what happens when you try to create big documents was unclear, and it was necessary to experiment a little to refresh my mind. On an 896KB system, a 288KB document was loaded, then merged on to the end of the memory copy, then part of it was merged on to the end again. The most that could be got into Quill was 217 pages, a file size of around 626KB. The DEF_TMP overflow file to flp2_ came close to filling a blank 720KB disk. The next question was, can a file of such size be edited successfully (accepting that almost anything was going to take a long time?) It was possible to type some more text in, and delete lines, but commands were refused, 'out of memory'. Effectively, there was an impasse. So, what happens if the overflow device is a hard disk? Much the same. The Merge operations had to be halted at the same

Continued on page 6

document length. The overflow file was, in fact, much smaller this time but it was clear the amount of disk space available didn't interest Quill; it had reached its limit. Funny things happen when a file this length is loaded in Quill, and you wouldn't sensibly want to trust what Quill would do to such a file. My recollection is that 350KB is about the limit Quill can handle sensibly. No, for serious work like this the capacity and speed of Perfection are a much better bet.

Reappearance

Welcome news for users is the expected reappearance later this year of a collection of well-known programs which disappeared from view about eighteen months or so ago. In the intervening time, most or all of these programs have been upgraded. It is a pleasant change for 'vapourware' to become real, for once, and to be coming from a reliable supplier.

Development of a new version of *text*⁸⁷ is well on its way to completion. A revised menu structure should make the transition from Quill to *text*⁸⁷ easier, but there will also be additional functionality for more experienced users.

My usual thought when reading about the wonders of the JS version of the QDOS rom is 'why bother?'. The JM has served me well for six years. The JS has benefits for

the programmer, but the user is not likely to notice them. However, serious troubles running revised versions of the two word-processing programs on my JM recently have been traced to incompatibility with the JM, and this may be a pointer to more trouble in future. However, Perfection in its latest, shipped version should run satisfactorily with the JM; the version of *text*⁸⁷ in question is intended for users of the Atari QDOS emulator with the 768-pixel screen driver, and that will normally (always?) have a JS anyway. Coupled with the trouble using one-third-height Mitsubishi drives, the rom factor may lead me to test programs on two different systems in future; one may have the JS rom, Gold Card, and (high-density) one-quarter-height drives, the other would have the JM rom, Trump Card, and one-third-height drives. My understanding is that there must be hundreds (maybe thousands) of QL systems in existence with both JM rom and one-third-height Mitsubishi drives (as supplied by Eidersoft); software and hardware developers ought to bear this in mind. Sounds a bit like the situation described in my earlier comments on of Minerva. . .

Readers' Letters

TK Computerware were good enough to call and suggest a way around the difficulty

R Thomson had experienced obtaining paper for his Serial 8056 printer. The suggestion is to contact a commercial stationery company (perhaps Ryman) and ask for details of thermal roll paper for fax machines. Make sure you specify thermal paper, as some fax machines use plain paper. A4 width (210mm) is normal, but some machines do use 216mm-wide paper (the Fujitsu dex 455/455, for example). There may be the usual (for the 8056) problem of paper not being quite the width of the original type; as this is a standard problem, it seems desirable to manufacture some type of shim(s) for the platen, to allow for narrower paper. A Quanta member, Richard Biggs, recently wrote (in the Quanta newsletter) that he has no difficulty obtaining 210mm-wide rolls (the 8056 normally takes 216mm?) and has indeed made an adapter to make up the difference. This paper costs him much less than the 210mm type, and he suggested he could supply three rolls and the adapter for £13. If Richard is a reader, and is prepared to offer this service to other 8056 users, perhaps he would let us know. Some Samsung fax machines are supplied complete with a spacer so that they will take both 210 and 216mm rolls.

The current, recommended version of Minerva is 1.90, so I am told. Any comments on software compatibility with this version?

JOCHEN MERZ SOFTWARE

Im stillen Winkel 12
4100 Duisburg 11 - W-Germany
Tel./Fax 0203/501274

NEW! Level 2 Floppy Disc and RAM disc driver for SuperQBoard and TrumpCard! New Emulator Software! QDesign for Laserprinters! QDOS Reference Manual! NEW!

QL-Emulator (Hardware, QJUMP's Drivers AtariDOS & utilities) £166
EPROM Cartridge for ST (makes machine autoBOOT), switchable £33
Centronics Cable for ST: 2m £6.50, 5m £10.50 **MIDI Cable 2m £3.50**
ST-Monitor-Cable: Scart £9, Cinch £11, Amiga 10845 £13
Floppy Adaptor to connect QL-Disk-drives to the ST £9

QDesign — A new graphics program which uses the Pointer Environment. Very easy to use, with extras never seen before in a QL graphics program: Editable area ranging 512x256 pixels up to 2880x2880 (depending on your RAM) You can even load monochrome 640x400 ST pictures and use them. Page View, Scaleable Vectorfonts etc. £38
LaserVersion for Laserjet II, Inkjet, Deskjet £45 **NEW!**

FLP/RAM Level 2 - Replacement EPROM for SuperQBoards (V1.17 onwards, with or without mouse) and all TrumpCards. No soldering! New Manual! Real sub-directories! QL - Emulator compatible! About twice as fast, with improved slave-block-handling! Backup - Dates and Version - Number, even accessible from SuperBASIC. £14 **NEW!**

QD III — Very comfortable text-editor, running under the Pointer Environment. Many features (e.g. Menu, Scrap) £38 Upgrade from VI or VII £10

QPTR — Re-released & Updated! Toolkit for Machine Code and BASIC Pointer Environment Programmers. Revised manual & keys/macros updated. CONFIG is also explained. £30 Update (new manual & disc) £14

QMENU — Many QD Users will know it: the Menu Extension. Now with more facilities, even real sub-directories are treated in a new file-select window. Programming instructions and examples for SuperBASIC and machine code. A Scrap Thing is also implemented now! £11

QPROMMER — Menu-driven, very comfortable EPROM-programmer software which allows you to use the ATARI Junior-Prommer under QDOS. £19

Package: QPrommer & EPROM-Manager II £29 **NEW!**

EASYTPR II — Create your own Pointer Environment menus & sprites and use them in your own SuperBASIC or m-code programs. Many users waited for this package, as creating Menus was not easy, until now! Put all the items you need on screen, that's it! Supports all the sub-windows and uses the Menu Extension! Now even complete application-sub-windows may be created! Many examples! £49

EASYTPR II — Budget version £29
Upgrade from version 1 £10

NEW! NEW!

QDOS Reference Manual — This book is a must for all m-code programmers. It explains how to use QDOS, all traps and vectors, the Thing System, the HOTKEY System II and much more. It points out which features work on a QL, an Emulator and how to write compatible for future operating systems. DIN A5, 170 pages. £30 Special Update Service available!

QSUP - System Utilities: translation-table editor, printer panel (now also for Star's) NotePad and much more £26

QLQ - 24-pin Printer Utilities with 13 download fonts, font editor etc. £21

Thing & EPROM Manager - Puts your files (Jobs, Things, Extensions etc.) together and creates files which may be burned into EPROM. £18.50

DIAMONDS — A "just one more go" game to keep you up at night. £11

Firebirds - Fast, difficult action game £10

Brain Smasher! - Mah-Jongg type game (best played with mouse) £12

Arcanoid II - Kick the brick! 40 Levels. £10

Ion Gold & Doppel Ion £10

Super Games Pack (Pengi, Arcanoid, Firebirds and Ions) £25

Pack of 10 3.5" Discs MF2DD incl. Labels £3.10

Keyboard Interface (for IBM-type PC/XT keyboards) £41

QIMI — QL Mouse Interface with Real-Time-Clock £41

High Quality Mouse for QIMI or ATARI ST £27

QIMI without RTC (for Gold Card) £37.50

Adaptor AMIGA mouse to ATARI or QIMI or v.v. £2.50

Please add £4.00 for postage and package.

All programs except Arcanoid, Firebirds & Ions need memory expansion. All prices excl. V.A.T E&OE

Cards Welcome!



QL

SCENE

July Quanta

The July 1991 issue of *Quanta* magazine (Volume 8, issue 6) contains reports on a number of projects, including the Qimi mouse which Quanta expects to be back in production, soon under its aegis; news of the library; a report on the recent London workshop; some elementary programming advice; the Quanta version of Dr Sohail Bhatti's Qlaw questionnaire, and various informed letters and short articles. Quanta membership is by subscription, full information from the Membership Secretary, Bill Newell, 213 Manor Road, Benfleet, Essex SS7 4JD.

Qimi from Merz and EEC

QL users have often been unhappy that they did not have a mouse readily available, particularly for graphics work. The main problem has been the lack of a suitable interface. The Qimi Mouse Interface is now being produced in Germany by Jochen Merz and distributed in the UK by EEC Ltd.

The mouse interface is assembled by opening the QL, removing the QL rom and the ZX8302 ic, inserting the Qimi pbc in the vacant sockets and replacing the rom and ic in the sockets on the Qimi board.

EEC informs us that QL users using an internal pc keyboard interface will only be able to in-

sert the Qimi interface and use the mouse if they mount the pc keyboard interface internally. The kit for doing this is available for £17 vat-inclusive.

The price of the mouse and interface is £75 vat-inclusive. The mouse, which is very smooth in operation, is £25 sold separately, and it works with the Atari ST and Amiga as well as the QL.

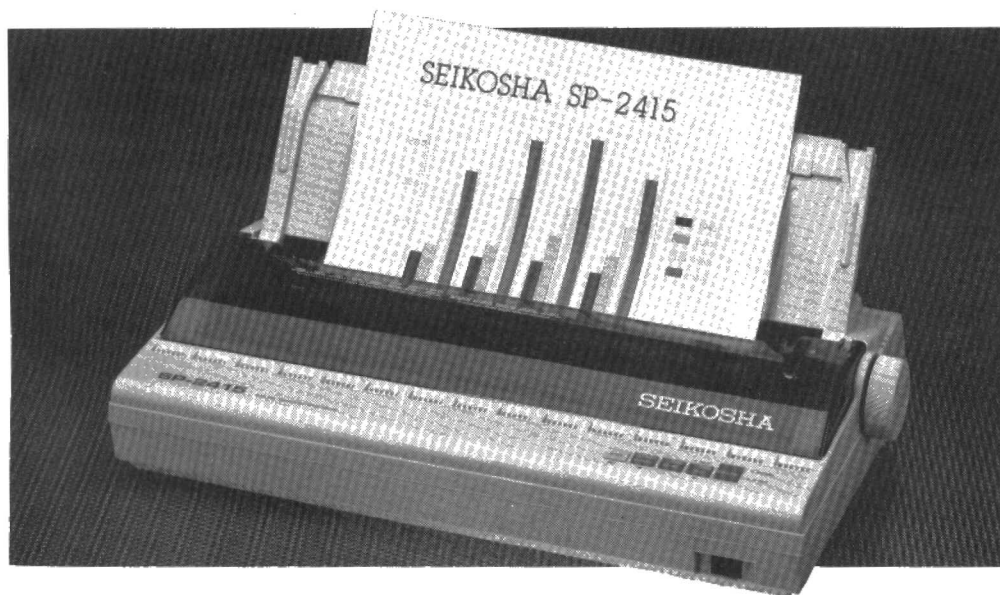
There is a considerable amount of software available for use with the Qimi mouse from Quanta, the public domain and other suppliers.

We have a correction to the item on Keyboard 90 in the

June 1991 issue: the price is £75 for the Interface, and the UK Keyboard is £35, all prices inclusive of VAT. The telephone number was also misprinted – please see below for the correct number.

EEC has asked us to pass on their apologies for delays in the supply of Keyboard interfaces and UK keyboards, and they assure us that stocks of all parts, including external mounting boxes, are now freely available.

Further information from EEC Ltd, 18-21 Misbourne House, Chiltern Hill, Chalfont St Peter, Bucks SL9 9UE. Tel: 0753 88866.



Heavy duty printer

Seikosha have issued the SP2415, a new 9 pin dot matrix printer described as a durable, wide-carriage machine capable of fast throughput, and suitable for heavy work-

loads. The maximum quoted print speed is 300 cps, with an average of 240/200 cps (12/10 cpi) in draft and 50 (10 cpi) in nlq. Noise level at top speed is 53 dB (A). The printer has stand-

ard parallel and serial interfaces, five fonts in a variety of print styles and type sizes and a variety of paper handling functions. The retail price is £320 plus vat. The printer is available through Seikosha's UK distributors. *Always check that printers are compatible with your QL interface before purchase.*

Book offer

Publishers Marland Press, are considering the possibility of publishing a book or books useful to QL users, if there is sufficient interest. Asking for people 'who can supply articles and program listings which would be of considerable help to other QL users', they wrote to *QL World*: 'The main aim would be to produce well-informed, sensibly-priced books, based on users' actual experience.'

They are looking for original material, not previously published, but many also be interested in reprinting agreements for 'any book considered of general interest but not currently in print.' They state that royalties will be paid on all material. The projected volumes would be A5 (Quanta-sized) format and comb-bound for easy use.

Interested users should contact **Marland Press at Lower Stone, Peters Marland, Torrington, North Devon. Tel: Langtree 258.**

OPEN CHANNEL

Open Channel is where you have the opportunity to voice your opinions in *Sinclair QL World*. Whether you want to ask for help with a technical problem, provide

somebody with the answer, or just sound off about something which bothers you, write to: Open Channel, Sinclair QL World, 116/120 Goswell Road, London EC1V 7QD.

Ctrl

Can somebody please tell me where I can get 'Ctrl' plugs from, how much they cost and what the 'pin' (if you can call them that) connections are? I am hoping to use some switches and self-written programs to allow disabled children from our youth club to enjoy playing on the QL.

In return I can supply a telephone number whence I recently purchased thermal printer paper. This is Abacus Business Forms. Tel: 0254 583653 — you had better ask them if they accept mail order.

In response to R Thompson, I am willing to copy a serial 8056 printer manual if he is willing

to pay for the copy and postage charges.

I have written a Mandelbrot program, however, it needs alterations to create a clearer picture. Amendments would be welcomed.

K Dunbar
Accrington
Lancs

Groups

Are you able to put me in touch, through the pages of the magazine, with any other users or user groups in this area? I would also like to know how to obtain different print faces on my Brother 1009. I know that it will do it, but the instruc-

Editor's Notebook

You will have noticed a subtle change in the way *QL World* is laid out this month. This is because we have moved to a different printer and our colour pages are arranged in a different order. Nothing more spectacular than that. We stopped running a colour centre spread after our last change of printer — some years ago now — and we won't be seeing that again while the *New User Guide* holds pride of place in the middle of the magazine. But we have some beautiful colour for you in Kevin Ball's article on Video Graphics on page 20.

Dr. Bhatti of QLAW has written an interim report on the response to the questionnaire run in *QL World* last month. So far the expected has happened: the two percent have come forward. This is normal for questionnaires in all fields.

QL Users, however, are not merely normal; they are persistent, independent and intelligent. So, if you are one of the 98%, don't be normal — send us your questionnaire.

```
5 WINDOW#1, 450, 220, 30, 10
6 z=10: REMark this number determines
  how many times the program goes through
  the calculations before it stops!
10 SCALE#1, 4, -2, -2
20 CoUnT=0
30 CLS
40 FOR y=0 TO 2 STEP (1.5/100)
50 FOR x=-2 TO 2 STEP (1.5/100)
60 e=0: d=0
70 CoUnT=0
80 a=(e*e)+x-(d*d)
90 b=2*e*d+y
100 CoUnT=CoUnT+1
110 IF CoUnT>z THEN GO TO 150
120 IF (a>=2 OR b>=2) THEN GO TO 150:END
  IF : REMark try putting and in place
  of or
130 e=a: d=b
140 GO TO 80
150 IF CoUnT<z THEN colour=CoUnT :END IF
153 IF colour=0 THEN LET colour=1
155 IF CoUnT>=z THEN colour=0:END IF
160 INK colour
170 POINT x, y: POINT x, -y
180 NEXT x
190 NEXT y
```

tion book might as well be in 'computerspeak' for all that I can extract from it.

David Conway
Wokingham
Berks

Agenda

The formatting problem Alec Wilson is having is probably twofold in nature. Firstly, the PrintSettings file in the Agenda must include the following:

```
BORDER = 0
ROLL
GAP-HEIGHT = 0
```

Note that any relevant Group Format file or Drawer Format file (if present) must also contain these instructions, otherwise they will be overridden. These instructions prevent the Agenda form inserting extra spaces which may be required when printing to a printer.

Note that the d drawer has a

built-in default Drawer Format file which is accessed by keying START d SHIFT SPACE.

The second problem is that when a group of files (eg diary files) is transferred by the normal printing method they are combined into a single file. It may be possible to overcome this using *Agenda Basic Code* as available from Microwriter Systems, but this is designed to run on a pc — although it will run under *Conqueror*. I have not tried using it for this purpose, however. You could transfer each file separately!

Microwriter Systems also sell a PC/Agenda comms program which is designed to back up the Agenda to a pc disk. This also runs to a degree under *Conqueror*, but so far I have not achieved a successful backup — I have put very little time into this as yet, though.

Graham Underwood
Workshop
Notts

Quill

I have stayed with *Quill* for several years. Now I have discovered *text87*, often mentioned in *QL World* because of its printer drivers, but not as the remarkable, all-round word processing program it is. I have gone over to it for my new Panasonic 24-pin printer.

Text87 is not effortless, and is often maddeningly awkward until suddenly a new aspect of this astonishing program is discovered. It has somewhat of a capacity for desktop publishing, but personally I go for its simple merits compared with *Quill*.

I have days, once every few weeks, when my QL locks up. I do not worry. I save repeatedly because it is so much easier with *text87* than with *Quill*. It saves with no questions asked and returns your cursor to where you left off. By saving to disk frequently, I cannot lose much on lock-up days.

Quill, an old friend, is awkward on Save, and scrolling is a pain.

While I have a version of *Quill* modified to print all the fonts built into my new printer, the fonts are not WYSIWYG. Page format becomes a guessing game. This is not so with *text87*.

I have yet to see *Perfection* in action, but one test for me will be whether it saves to disk as effortlessly as *text87* for use on those tricky QL lock-up days.

Stanley Horwitz
Cirencester

Editor's comment: everyone has his or her 'druthers. DP have demonstrated Perfection at a number of user-group meetings and shows round the country — if you want to see Perfection, keep your eye open for one of those.

Q Jump

The *QJump* Extended Environment is the *only* program for the QL, and it is *not* biased towards hackers. The mouse, which must be an integral part of the computer, in terms of both software and hardware, already exists, and is called *QIMI*.

The concept of directories on the QL was introduced in

ToolKit (or was it *Toolkit* 2? I don't remember, but anyone using the QL for any kind of programming should have *ToolKit2*). The difficulties arise when software-houses fail to make their software configurable in a decent way, with regard to where they read their files from, etc. Many programs, including *Quill*, can only load from MDV1_ or FLP1_ or any such device name, limited to five characters in length. The intelligent programs, such as *text87*, can be configured to load their files from a device-name with any length, such as WIN1_text87files_. Once such a program has been configured, and its files moved to the relevant directory, with a single command such as:

```
WCOPY          FLP1_
WIN1_TEXT87FILES_
```

it can easily coexist with other decent programs, without the need of any further commands.

Other programs, although not DATAD\$ (which is set with DATA_USE). The programs can easily be used, but not multitasked with other such programs, since DATAD\$ cannot have two values at the same time.

The command Bryan Davies refers to as a 'change-directory command' could be WIN2, a 'patch' that allows programs such as *Quill* to run, using subdirectories. After a command such as 'WIN2 Psion_', references to win2_file_doc are (invisibly to *Quill*) transposed into references to win1_Psion_file_doc. Voila, *Quill* runs from within a subdirectory, called 'Psion_1'. This command cannot handle more than one program at a time, but by now most people use newer, better programs (such as *text87*), which can be configured easily. Don't they?

Commands to change the current directory are mentioned in the *ToolKit2* manual: DDOWN, DUP, DNEXT and DLIST.

If you need to know the current defaults, as when using an IBM-PC which usually integrates it in the prompt, compile this program:

```
100 REP L:AT 0,0: DLIST
```

and EXEC it.

Please, anyone interested in

REAL C (not Small C), please contact me. If you are a legitimate owner of MetaComCoC, I will supply you with my own updates, with these features: runs without ROM, uses the DATA_USE-default, so it can indeed run from a subdirectory, from any media, the memory- and I/O-library has been completely rewritten, produces ROMable code, if the code is 64K or less. Please send 1 disk, and IRCs to cover expenses, when I mail the disk back to you.

Erling Jacobsen
Lyngby
Denmark

ProPub

I recently received from Digital Precision a copy of their upgrade of *Professional Publisher*. It is much quicker to load than its predecessor and the program itself works faster. For those who are interested in some facts and figures, the loading time on my JS Trump Card machine without *Lightning* (including the pause for TK2 Ext) is reduced from 2 minutes 15 seconds to 55 seconds using the full program as supplied by DP.

In addition, I got the impression that the program itself was slicker and running faster. I queried this with Freddy Vaccha of DP, who told me that the program did indeed run 2.5 to 3 times faster. The increase in program and loading speed had been achieved by including Runtime_exts, eye_q_exts and mouse_exts within one file, DTP text.

At times I prefer to have extra money available for other tasks, so I reconfigure PP to reduce the page and font memories and increase the page width pixels to suit myself and my

printer. The loading time for my on is now 40 seconds, which is comparable with *Page Designer 2* with a similar configuration.

I am very pleased with the upgrade. Every time I use the program I am impressed with its versatility and continue to find new and useful features. *The New User Guide* in the March 1991 *QL World* stated (paraphrase) that the extent of the user's imagination is often exhausted long before the limits of the computer's physical capabilities, and the logical constraints of Qdos are reached.

Professional Publisher is a program which follows this philosophy and allows the user complete freedom of imagination in any page that he constructs. Constraints are most likely to be due to the operator's imagination rather than the program.

Peter Hamill
Elton
Peterborough

PS For those who are interested and have *ToolKit 2*, I have an alternative to the PP manual advice to move the three auxiliary files to another disk to make room for something like *Lightning*. If you keep *Lightning* on a separate disk, and alter its boot program as suggested below, you will find that once *Lightning* is booted, you can remove the disk, put in PP or whatever you want, and by pressing ALT b automatically boot whatever file is in Flip1_.

I find it useful to keep this *Lightning* boot on a separate disk, especially when I want to try out new programs, or for general use, as it saves repetition of the *Lightning* BOOT program on each disk.

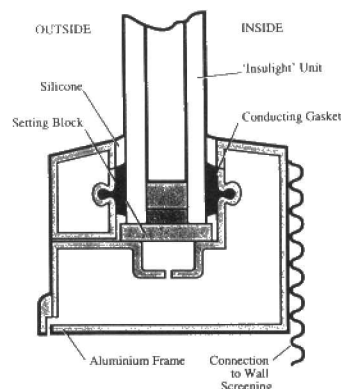
```
10 ReMark - CONFIGURED LIGHTNING BOOT FILE
20 REMark Do NOT use with other configured Lightning
systems.
30 source$ = 'FLP1_'
40 S=RESPR (33280)
50 LBYTES source$&&' lng_TEXT_ext', s+0: CALL s: PAUSE
60
60 LBYTES source$&&' lng_GRAF_ext', s+21758: CALL
s+21758: PAUSE 60
70 LBYTES source$&&' lng_MATH_ext', s+26016: CALL
s+206016: PAUSE 60
80 TK2_EXT
90 if$+CHR$ (10)
100 ALTKEY 'b', 'lrun flp1_boot' &lf$
110 NEW
```


'Data-proof' glass development

Datastop is the name of a new 'spy-proof' glass developed by glass giants Pilkingtons.

Computer data can be stolen – or simply received – through the medium of electromagnetic radiation, carrying signals equivalent to the data being handled by the computer, escaping from the computer room through ordinary glass windows, where they can be picked up by certain types or receiving equipment. Receiving these signals has become something of a growth sector in the industrial espionage business, so it is no wonder that the Ministry of Defence, the Foreign Office and institutions in the City of London are among those who are "already showing interest", according to Pilkingtons.

There is also the danger of computer data being corrupted



by unwanted emr – radar, for example – coming from outside the computer room.

Older security systems often called for rooms which were either windowless or had small, wire-mesh-covered windows.

Datastop glass reflected electromagnetic radiation from inside and outside the room by means of a special coating on a double glass sheet.

Tests showed that an office glazed with Datastop was electromagnetically secure "even when a receiving aerial was sited immediately outside the window."

For maximum shielding, the glass pane is connected by electrically-conducting gaskets to the metal window frame, which is in turn earthed to screening material in the walls.

The great advantage of this glass is that secure computer rooms can have the same advantages of view and daylight (and access to fresh air, presumably, when the systems are not running) as non-secure offices.

Italiana

Two Italian QL clubs, QLItaly Group and Club-Ware have decided that 'they were doing much the same thing, and so they have joined together' in January 1991 to form Qitaly Club, the Italian club for QL Users.

Qitaly is publishing a public domain disk magazine, *Qitaly Magazine* — the disk can be freely copied and distributed to non-members. The subscription to Qitaly is 20,000 Lira a year, with an additional 30,000 Lira if you want six issues to Qitaly Magazine. Single copies of the disk magazine are available for £3 (cash or international postal order, to five International Postal Reply Coupons (IRCs) in UK prices, or the local currency equivalent for users ordering from other countries).

The Chairman of Qitaly Club is Roberto Orlandi, Via Brescia 26, 25039 Travagliato (BS), Italy. Tel: (local) +39 30 6863311. The editor of *Qitaly Magazine* is Dr Eros Forenzi, Via Valeriana 44, 23010 Berbenno (SO), Italy. Tel: (local) +39 342 492323.

The Club is making plans to organise a Third Italian QL Users' meeting in September or October in Turin. The dates are not known at the time of going to press.

Warning

Epson UK has issued a warning that Epson-branded consumables manufactured in Canada do not meet the specification standards set by the Seiko-Epson Corporation.

Printer ribbons are specifically named, and Epson (UK) lists shortfalls in pull-out tension and rewind torque on the ribbon mechanisms, weave and join construction of, in particular, 24-pin print ribbons, and ink formulation. Erratic print quality, variable print yields, ribbon jam, printhead failure and even motor drive burnout are listed as possible faults arising from the use of non-approved ribbons. Seiko-Epson's approved ribbons are manufactured in the UK, Japan or Germany.



Star back green award

Boys and girls from the 96-strong 1st Ebbw Vale Peniel Scout Group with help from the Prince of Wales' Committee, took over a derelict site and transformed it, planting flowers and three hundred trees, as part of the preparations for the town's 1992 Garden Festival.

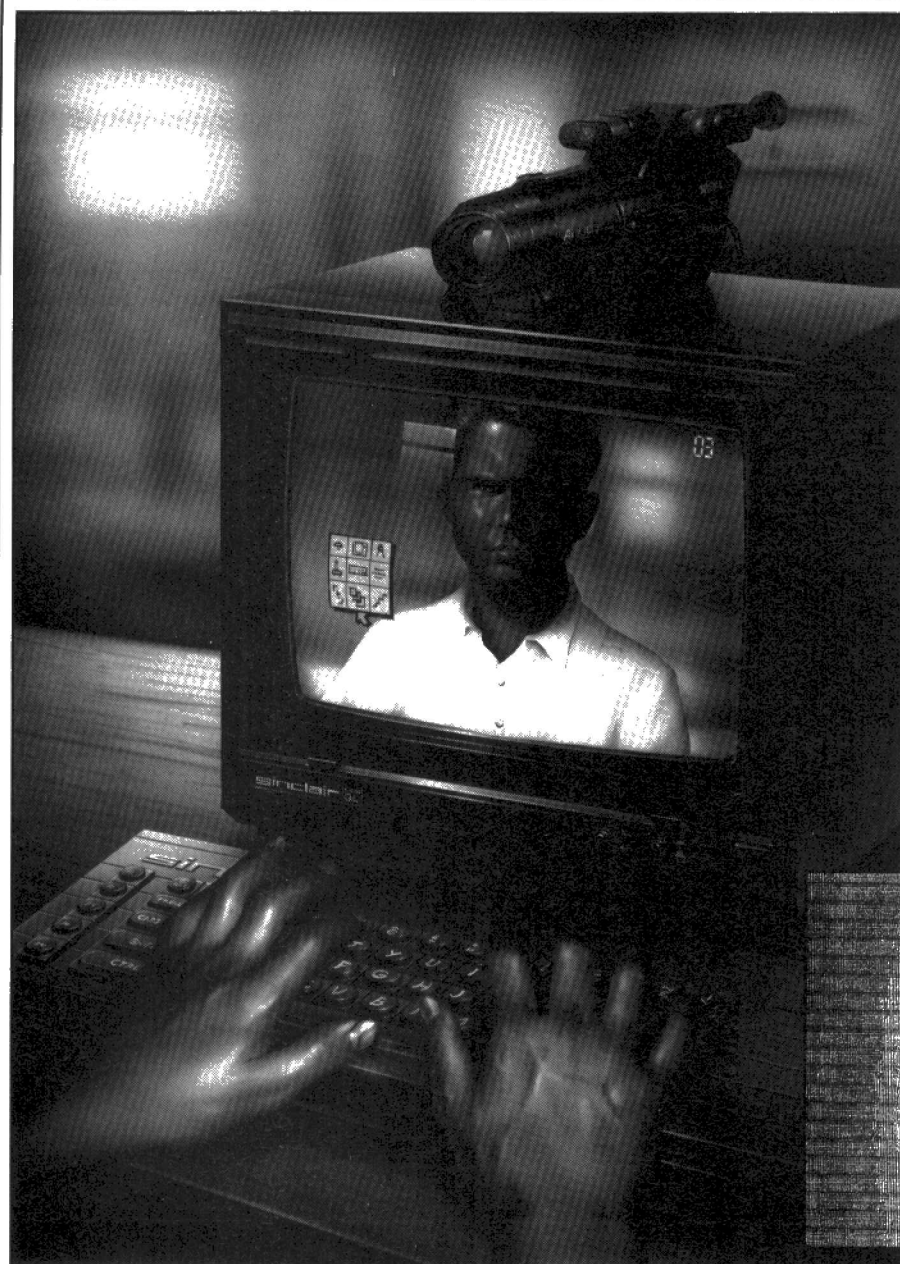
Printer makers Star Micronics who have a manufacturing plant in Tredegar, South Wales, donated their millionth printer to the Prince's cause.

Scout Alice Griffiths, 12, and senior patrol leader Jonathon Williams, 15, received the millionth printer on behalf of

the Prince of Wales' committee from Star's sales and marketing director Roger Bayley.

The land reclaimed will raise money, serving as a camp site for visiting scout troops during the Garden Festival, starting in May 1992.

Troop Leader Cllr Bill Pritchard took the opportunity to launch an £1100 appeal to raise money to help bring piped water to the site.



The QL may not seem an obvious first choice when high quality graphic work is required, but with recent hardware developments and good quality software I've found that excellent results can be achieved, albeit with some effort. Over the last six months I've been involved in producing a variety of QL graphic screens which I have then put in sequence and recorded onto video tape for titles or for promotional videos.

As I do not consider myself to be very artistic, I soon realised that better results could be obtained by using a digitiser. Originally I used the Spem digitiser. While this produced good results, I have found that the digitiser which I currently use, manufactured by CL Systems, is ideal for my purposes. That a machine like the QL can have a choice of two digitisers would seem to indicate that there is life in the old QL yet. A digitiser, as you may know, converts a video image into a screen format recognised by the QL, and capable of being imported into desktop publishing applications or saved as a 32K QL screen file.

In addition to digitised images, I needed a good quality graphics program to edit the digitised screens and to add text. As my QL set-up has a Qimi mouse interface fitted, and as I like the pointer environment,



A SPEM digitiser demo screen

VIDEO GRAPHICS ON THE QL

I There are now a number of items of software and hardware for the QL which enable the capture and editing of video graphics, Kevin Ball has built up his own system and describes what he has been able to achieve with it.

I chose the *Painter* program, the version I'm using is V3.04, although the current version is now V4.01. I believe that a version is also produced for the Atari ST/QL emulator.

The digitiser really is easy to couple to the QL. All that is needed is to connect the fairly large digitiser to the rear rom socket of the QL making sure that the power is off. It helps if the three plastic legs of the QL are removed also as the QL and digitiser can then lie flat on the desk. Having reconnected the power supply, to get things going you have to boot up the software provided. In my case this is on disk as I use a trump card and twin disk drive system, but the digitiser manufacturers state that it works on an unexpanded QL system too, and they can provide the software on microdrive.

A SPEM digitiser demo screen

Connection from a video source to the digitiser requires a lead to be connected from the video out socket of your video machine to the back of the digitiser using a phono plug. If your video machine uses scart connections then you will need to obtain a lead which has a scart connector on one end and phono and DIN fittings on the other. The two phono plugs are likely to be marked video in and video out and the DIN should be marked audio. It is only necessary to connect the video out phono to the digitiser. Suitable leads are available from some high street stores or possibly Care Electronics. Your monitor or television should be connected in the usual way direct to the QL.

Using the digitiser couldn't be easier – it is fantastic to watch. The first difference from the Spem digitiser is that I could use my Microvitec Colour Monitor to view the digitised image. This wasn't possible with

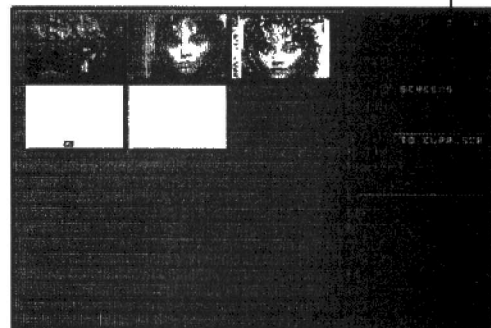


Three images compiled into one screen with *The Painter*

Spem and I had to use a monochrome monitor, although I understand that some versions of Microvitec work with it. The CL digitiser also allows a television to be used as a monitor – connected to the UHF socket of the QL. Again, this isn't possible with the Spem digitiser which requires an rgb connection. In addition to the monitor or television on which to view the digitised images you will need a television to view the normal video image – this should usually be connected to the RF out socket of the video recorder. This may not be necessary if you are using a video camera to provide material for digitising, as you will obviously be able to see what you are aiming at!

The CL digitiser operates in real time converting video images into QL screen format. At any point a particular image can be grabbed and then saved onto disk or microdrive as a standard 32K screen file. There is only one hardware control on the digitiser – a black level control situated at

the rear of the unit. This allows a considerable degree of control over the contrast level of the digitised image. The digitiser has three screen modes – full screen, quarter screen, or four quarter-size screens showing simultaneously. It operates in mode 8 and the screen is updated at the rate of five frames per second in quarter screen mode, or just under three frames per second in full screen mode. The fast digitising rate – 0.2 second is a real boon, because it means images can



A view of *The Painter* with twelve screen in memory



Five guide screens

be grabbed as the video plays back normally – in contrast the Spem digitiser takes about five seconds to produce a digitised image and so requires the video to be paused. I've found this sometimes produces interference lines on the image, as well as possibly causing damage to the video heads. However, the Spem digitiser has more features in its software which include a program to view the screens, recolour them, edit them or produce print-outs of them.

I tend to work through the video material saving suitable images onto one or more disks. Having finished the digitising stage of the process I load the *Painter* program and then compose a new screen using parts of one or more of the digitised images saved earlier. Many people will be familiar with the *Painter* program which has been reviewed in *QL World* and provides a host of features. Most useful for me in this context is the ability to have up to 12 screens in memory at any one time (upgrade memory permitting)! Images can

be manipulated between screens, enlarged, reduced, recoloured or turned around. One screen can be used to paint parts of another and text of all sizes can be added. Given these facilities and many more, I wonder why my end result is sometimes so mediocre!

Because of the many features of the Painter program I did find it complex to use initially – but it is worth persevering with, and I will be upgrading to version 4.01. I have found some bugs in the program with my set-up – many of the features would not work with one of my QLs which is fitted with a Minerva rom, and I understand problems had been reported with various QL roms. There is a particularly nasty bug which occurs when I multi-task Painter on my JM QL using Qpac2 and attempt to scroll part of the screen – my work of art usually disappears in front of my eyes! I hasten to add that the program manufacturers believe that their current version is compatible with everything – now that's a proud boast and one which I hope is correct! I tend to save digitised screens with the `_dig` suffix, and completed screens with the `_scr` suffix. this convention can save considerable time when wading through five or six disks looking for a particular sequence of screens.

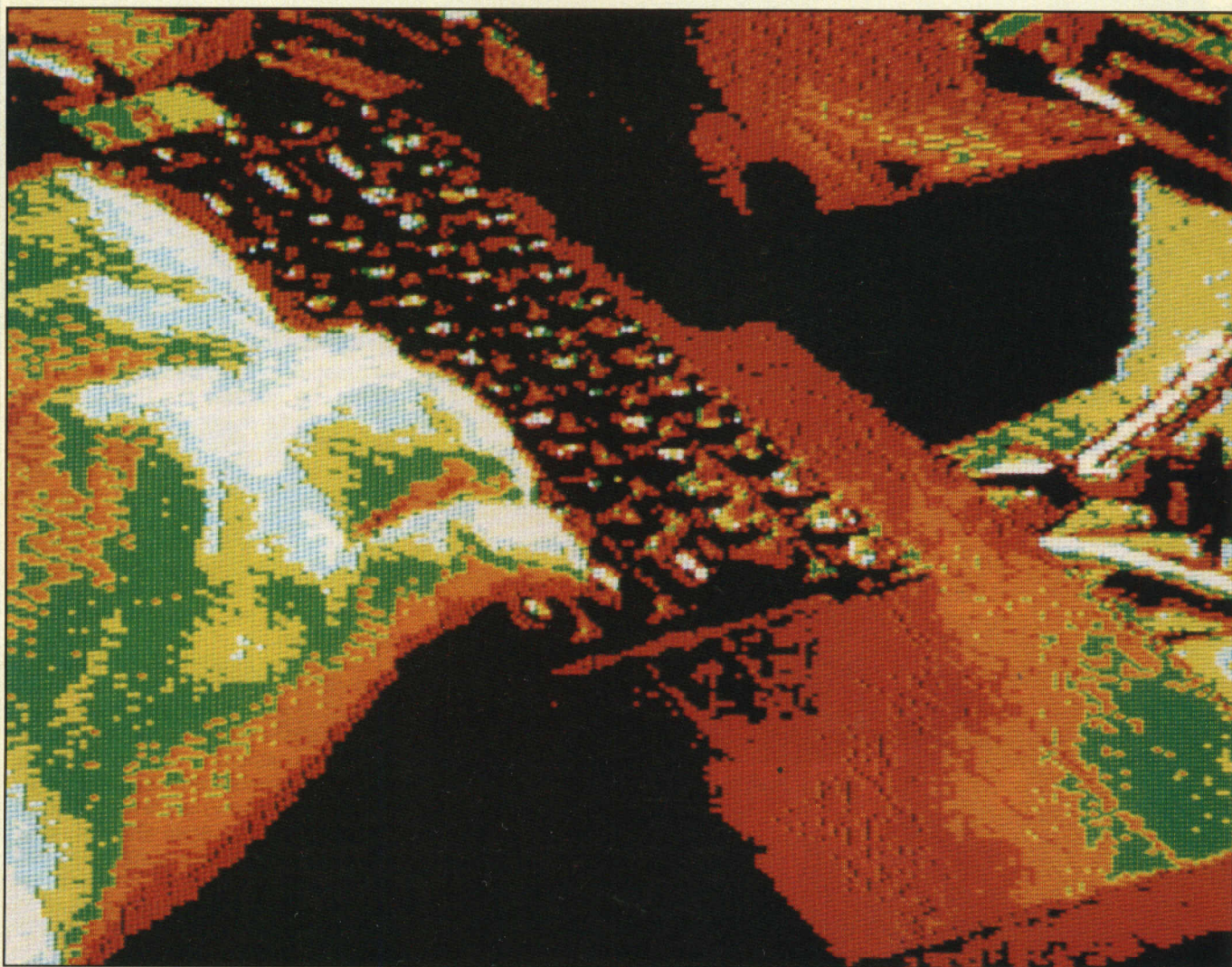
Having created a number of screens which I want to record onto videotape, I turn to yet another important QL program – *Vision Mixer*. This was reviewed in QL World in January 1991, and for anyone interested in graphic presentation I believe it is absolutely essential and excellent value at only £10. I understand that an even more comprehensive program has been developed by the suppliers (*Vision Mixer Plus*) but I haven't seen this demonstrated as yet.

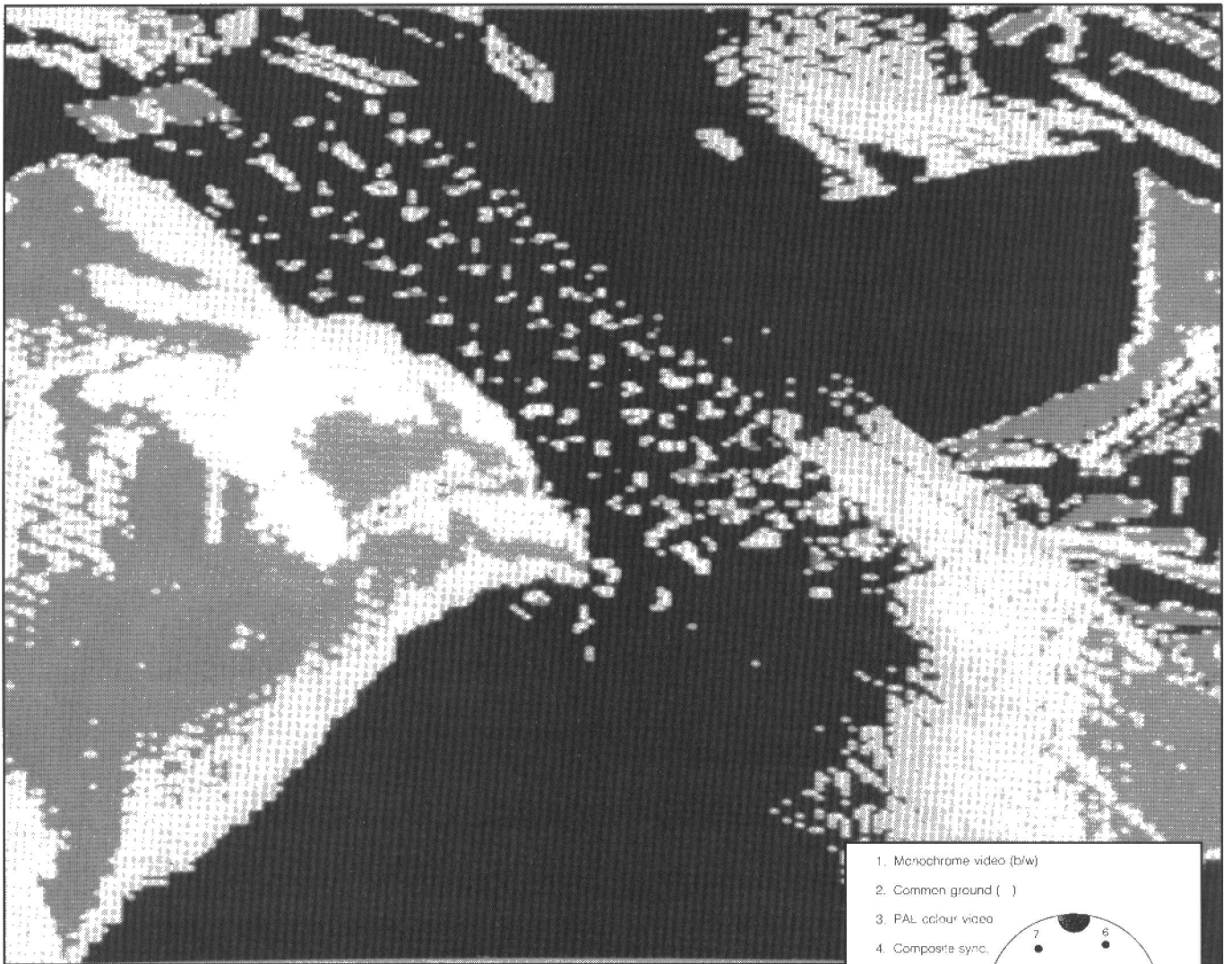
Vision Mixer allows me to load 21 screens into memory (using a 768K trump card) and then display them sequentially or at random using an incredible range of video effects. It is possible to have each screen displayed for a different time period or a random time limit within parameters set by the user. The effects used to change from one screen to another can be chosen by the user from over one hundred. The program requires one blank screen for some of the effects and the colour of this can be chosen by the user. I've found that when I want to record a sequence of text titles only, perhaps to be used at the end of a video, that it is helpful to select the colour of the blank screen to be the same as the background colour of the title screens. I then use only one video effect – random lines – and the sequence of

titles appear and disappear smoothly, giving a very professional effect.

For many people the ability to show screens on their QL perhaps as a shop window advertising medium will be all that they require, and the foregoing illustrates how well suited the QL is for this type of work. However, for my uses it was necessary to record my work onto videotape which meant that the finished results could be used by anyone who knew how to work a video recorder, and perhaps more importantly no computer expertise was required on his or her part, (and my QL wasn't at risk of being damaged!) There are other advantages in videoing – it allows any number of screens to be included in the sequence – by recording in stages – and it allows a soundtrack to be added, either at the time, or using the audio dub facilities of the video recorder at a later stage. The disadvantage is that there is some loss of quality – and if the screens are being designed on a monitor, it is important to ensure that designs don't go too near the edge, otherwise when the recorded version is shown on a television some of the message may be lost.

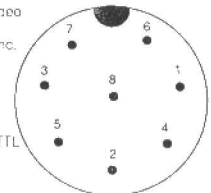
The actual process of recording can, initially at least, be torturous. With the power off you will need to connect a lead from the RGB socket of the QL to the





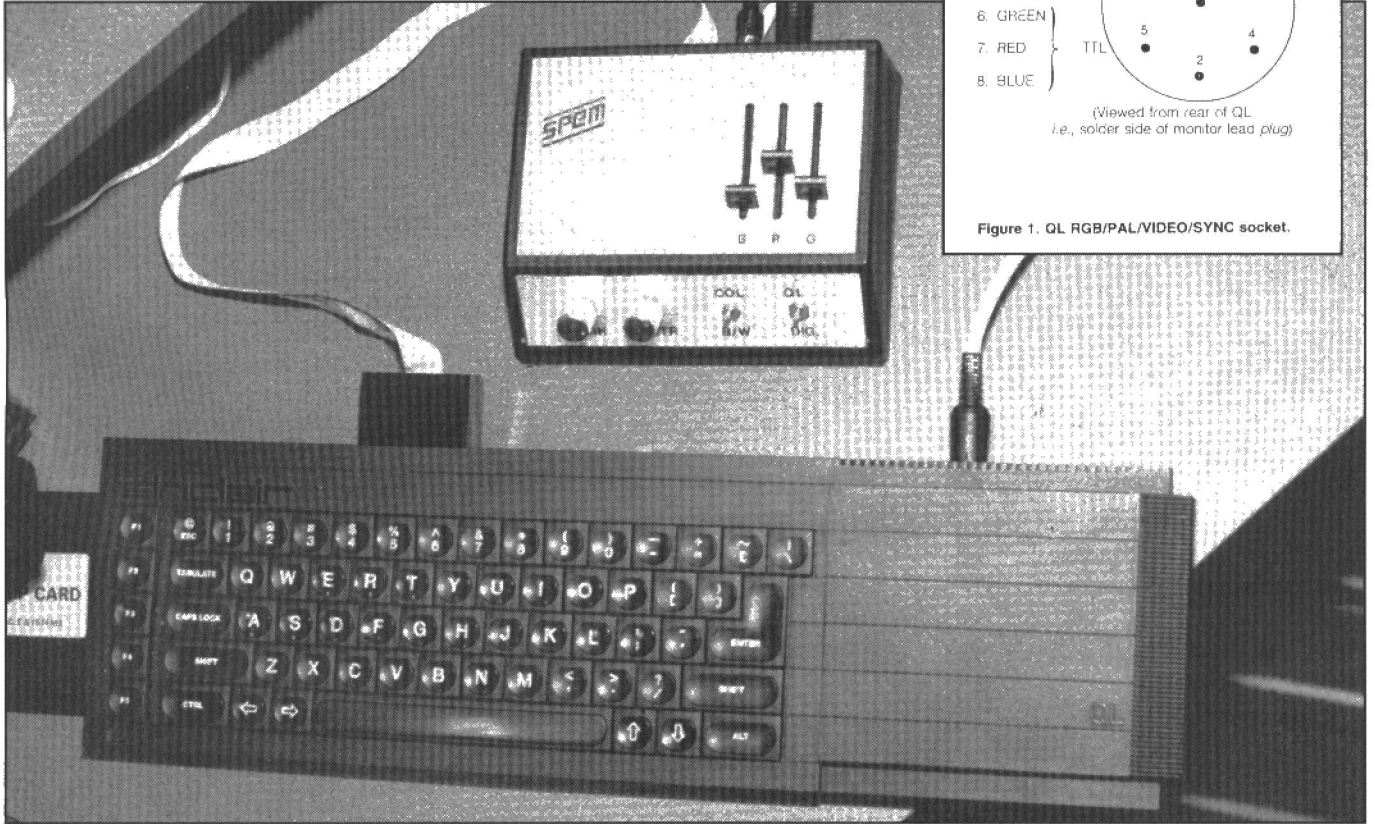
Above and left: mode 4 and mode 8 digitising. Below: the SPEM digitiser connected.

1. Monochrome video (b/w)
2. Common ground ()
3. PAL colour video
4. Composite sync.
5. — (nc)
6. GREEN
7. RED
8. BLUE

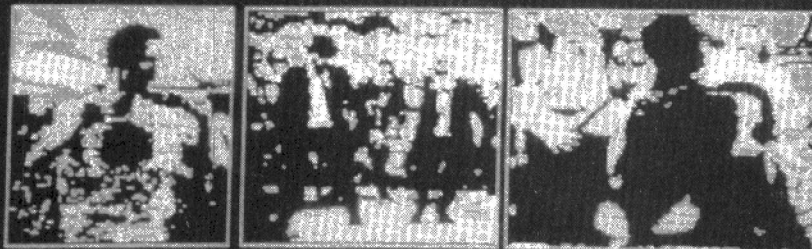


(Viewed from rear of QL
i.e., solder side of monitor lead plug)

Figure 1. QL RGB/PAL/VIDEO/SYNC socket.



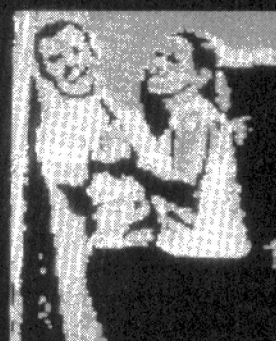
HALIFAX PROBATION PRESENTS.



TOP OF THE POPS!

Images compiled the The Painter

DANCIN
DANCIN
DANCIN
DANCIN
DANCIN



video in socket of the video recorder. It is likely that you may need to get a lead specially made for your machine. The one I have terminates in four phono plugs thus allowing access to a variety of pin combinations. If you buy a similar lead be careful about experimenting because it is easy to blow the chip in the QL. Some versions of the QL user guide giving details of the pin connections were inaccurate, but I understand that the diagram printed here from QL World Technical Helpline 1987 is accurate.

Once the QL has successfully been coupled to the video recorder, you will need to connect a television once again to the video, probably via the RF out socket. This will act as monitor for the QL. All that remains is to power up the QL, load vision mixer and the screens you wish to record and then begin recording.

With all of the electrical equipment around it is possible that you experience some interference - I've found that fine tuning of the video machine's RF select control can improve picture quality.

I recently needed to obtain a *Top of the Pops* type of effect to act as an introduction to a video showing staff at my office singing and dancing (for charity). I wanted to retain the original soundtrack and so I used two videos - one provided input to the digitiser while the other recorded output via the rgb socket of the QL. I simply used the digitising software to obtain moving digitised images which were immediately recorded onto the second video. I could even act as producer by switching screen display modes between full and

quarter displays. I attached a lead directly between the audio out socket of one video machine to the audio in socket of the other and so re-recorded the original soundtrack!

There is little doubt that graphic work is a developing area in computer use and it is refreshing to see that the QL can produce results that out perform newer and costlier machines. The choices of software and hardware that are available to assist in the various tasks put the machine into a class of its own. With video cameras being available for hire for around £15 per day it is relatively easy to produce personal digitised images to use in the way described above or for inclusion in desk top applications.



The CL digitiser connected to the QL's rom socket.

INFORMATION

Digitisers:

CL Systems
403 Chapter Rd
Dollis Hill
London NW2 5NG
Tel: 081 459 1351
Price: £123.00

Spem

via Aosta 86
10154 Torino
Italy
Tel: +39 11 857924
Fax: +39 11 280009
Price: £120.00 + vat
The Painter:

Progs

Haachtstraat 92
B- 3020 Veitem
Belgium
Tel: 016 48 89 52
Price: £50 (BEF 3000)

Vision Mixer

Vision Mixer Plus:

Dilwyn Jones Computing

41 Emrys
Tal-y-Bont
Bangor
Gwynedd LL57 3YT
Tel: 0248 354023
Price: £10
Plus Version: £22.50

SOFTWARE FILE

INFORMATION

Program: *QTop* (for QL and Thor computers). V1.00

Supplier: Cowo Electronic,
Munsterstrasse 4, CH-6210, Sursee,
Switzerland.

Price: £35 disk, £39 mdv.

Cowo Electronic has sent *QL World* a copy of its Qdos user interface *Qtop* for review on an Atari/QL with hard disk. The package is sold through TK Computerware in this country, and Qlympic Computer Systems in Germany and costs £35 on disk or £39 on microdrive. The centrepiece of *Qtop* is a utility called *Tdesk*, and I shall use *Qtop*



I Hugh de Saram runs Cowo *Qtop*, originally designed for the Thor, on a QL emulator and on an extended QL system.

Qtop

and *Tdesk* more or less interchangeably in what follows.

The system used for the review consisted of an Atari Mega2 with Extended4 QL Emulator by Jochen Merz, a 40 MB slimline scsi hard disk from Power Computing, a Philips PRO 8CM852 colour monitor and a Canon Bubblejet BJ-300 printer.

The first thing that struck me was the contrast between the claims in the advertising blurb that the system was written in optimised machine code and the front screen of *Tdesk*, which announces that it is in Turbo-compiled SuperBasic. Perhaps we are supposed to assume that Turbo-compiled SuperBasic is optimised machine code, but it sounded slightly strange.

I started by copying the contents of the master disk into a new directory on my winchester – *win1_cowo_*. I decided to call it. The master disk contains two sets of programs, the spare set all prefixed with the word *SPARE_*. So barring the

most drastic of accidents, you should always be able to resurrect a copy of a deleted file. The disk arrived write-protected anyway, so the chances of making a mistake seem remote.

I then had a careful read of the manual, and wrote a little boot file, as follows (the *win1_cowo_* in line 100 was crucial for making *Tdesk* look for its ancillary files in the *win1_cowo_* directory. It will not function at all without the files *Tdesk_Opt*, *Tdesk_Txt*, and *Tdesk_Fnt*):

```
10 LRESPR win1_cowo_thorst_bin :
REMARK required by Tdesk
20 LRESPR win1_atari_bin
30 LRESPR win1_tools_winch_rext
40 LRESPR win1_qram_ptr_gen
50 LRESPR win1_qram_wman
60 LRESPR win1_atari_ataridos_rext
70 LRESPR win1_qtyp_spell
80 LRESPR win1_tools_qload_bin
90 HOT_GO
100 EX win1_cowo_tdesk; 'win1_cowo_'
```

Actually, I tell a lie. This is what I *should* have done. But being impatient, I skimmed very imperfectly through the manual, and wasted a lot of time trying fruitlessly to get the program to work by trial and error. So do read the manual carefully, it will save you time in the long run. It is well-enough written, and reached me as a loose-leaf A4 file, probably printed on a laser, although a couple of the illustrations appear to come from a dot-matrix screen dump.

Once loaded up, there is a delay of what seems like several seconds before the *Tdesk* screen appears, and it occurred to me that it would have been nice if there had been a Loading: please wait . . . message on the screen meanwhile, since I was left wondering for a moment whether something had gone wrong.

Qtop claims to have twice the power of *Qpac2*, so it is against this package that I shall judge it. I use *Qpac2* as a matter of course, together with elements of its older sibling *Qpac1*, and although it took a bit of experimentation to set it up and get the

most out of it, I find it a really excellent system to use – as indeed one might expect of a second-generation product from Tony Tebby.

A major item in Qpac2 is its housekeeping – my shorthand for the facilitating of filing activities on magnetic media. For example, it will display the file information for a particular disk or sub-directory in a window which adjusts itself in size depending on the number of files in the directory, and allows you complete freedom to resize it if you wish. You can then motor around the window at will with the mouse or the cursor-keys.

One file

By contrast, Qtop only allows you a view of file-info one file at a time; in its own order of choosing if you select ALL, with no going back if you go past the file you want; through a window of its own sizing; and once only before flipping back automatically to the previous menu. On the other hand, for a single individual file, it was quick and easy.

The thing I really missed here was the option to order the files as I wanted them, particularly by reverse date and time, so that I could back up only the most recently altered or created. Backup is a major factor for hard-disk owners, and to do it sensibly you must be able to order the files in the manner mentioned and put that info on the screen for all files at once. Of course, a dedicated backup program might obviate the need for this.

Tdesk scores well, however, on its display of sub-directories. You click on win1_, getting a display of everything in the top directory, and then simply click on a directory name to get a display of the files in that directory. This is simpler than Qpac2's

arrangement, and worked very smoothly and swiftly.

Although the manual implies that Qtop works happily without the Qjump window-manager (ptr_gen and wman), I found that it then caused weird things to happen with a couple of the most commonly-used SuperBasic functions.

Both dir and wcopy seemed odd: I kept finding that I was being switched to the Tdesk job, even though I thought I was working perfectly normally in SuperBasic. It seemed to be connected with using wcopy and being queried about copying and overwriting files: pressing Y caused a switch of jobs! A similar job-switching side-effect occurred with dir.

A second annoying thing is that the normal SuperBasic screen at channel 0, where you type SuperBasic commands,

“I would say that the ptr_gen and wman files from Qjump are essential to operation . . . they are supplied.”

was partly overlaid by the clock display of Tdesk, and since this is updated every second it is no good trying to solve the problem with cls – you just get a refreshed clock display!

I would say, therefore, that the ptr_gen and wman files from Qjump are absolutely essential to the satisfactory operation of Qtop. Fortunately, they are supplied on the Qtop master disk, together with Hot_Rext which enables you to program the keyboard and set up hot-keys.

Apart from general filing, the other major function of Qpac2 and Qtop would seem to be job control: starting, removing and switching jobs, or tasks, and controlling their screens and use of memory. Both Qtop and Qpac2 do this well, and there is little to choose between them in this re-

spect, except that it wasn't obvious how to decree the amount of memory to be used by the Psion quartet under Qtop (although a similar facility for Xchange is built in, reflecting the THOR bias of Qtop), whereas that is clearly explained under Qpac2.

Access

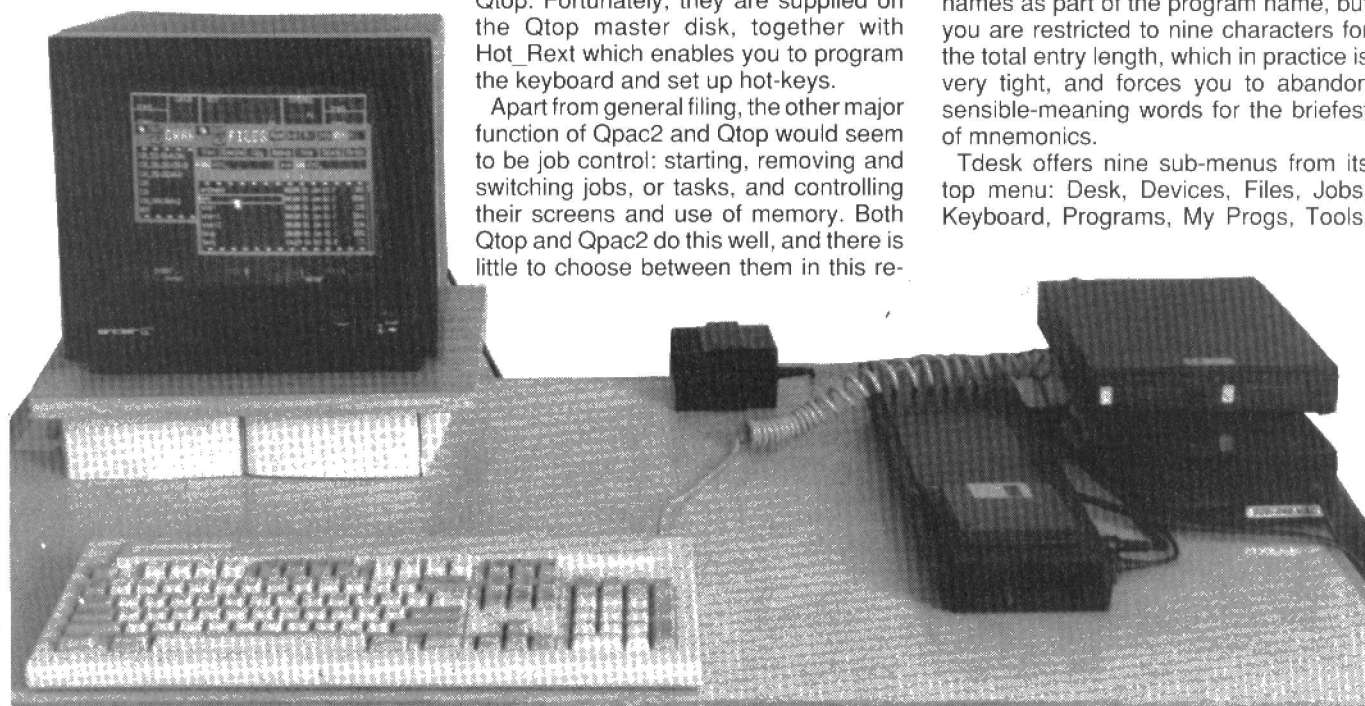
Qpac2 gives you, in addition, full access to every channel opened by the operating system. I find this facility particularly useful when something I have sent to the printer refuses to be printed. It is usually a result of leaving the channel open from some other task, and access to the channel table shows me immediately where I stand. However, as far as I could see, this important function is not available under Qtop, and must count against its claim to be twice as powerful as Qpac2.

Qtop allows you to write eleven program names into a menu called My Progs, and save this menu as part of your own private configuration of the Qtop package. You can then click on any of these programs with your mouse button, and it will load. However, unless you are prepared to change the default directory before loading each one, they all have to live in the same directory on your hard disk (designated the TD-Device and set via the Devices sub-menu – see below), which I found frustrating. I keep all files associated with *The Editor* in a sub-directory called edit_, all files associated with Text⁸⁷ in a sub-directory called text_, and so on. In this respect, therefore, Tdesk seemed a bit restrictive.

One file

It is possible to set the TD-Device to win1_, and then include the sub-directory names as part of the program name, but you are restricted to nine characters for the total entry length, which in practice is very tight, and forces you to abandon sensible-meaning words for the briefest of mnemonics.

Tdesk offers nine sub-menus from its top menu: Desk, Devices, Files, Jobs, Keyboard, Programs, My Progs, Tools,



Options.

Desk offers you a Help facility; information on memory use, system setup, colours available, Prog Info (which turns out to be the Tdesk front screen – rather a waste of space!); control of the network file-server and access to a number of clocks in fancy formats, and several demo programs, which will no doubt be of interest to some.

Defaults

Devices enables you to set defaults for programs, data, spooler, flp_use, win_use, ram_use, and for Tdesk itself. This is convenient and functional.

Files and Jobs speak for themselves and have been covered already. Keyboard enables you to set the language for your keyboard, which is a big selling point of the Thor, but no use for the Atari as far as I am aware. Programs is a fixed menu enabling you to start up some of the QL's favourite programs such as *Quill*, *Archive*, *Xchange*, etc.

However, if you don't happen to use these programs, or if you use them with different names as I do (in order to remind me how much memory I have assigned to the Psion programs with the Grabber facility, I tack a number on to the program name – *Quill50*, *Arch100*), then they are not very useful. As with *My Progs*, they all have to reside in the TD default directory.

My Progs I have talked about. Tools enables you to do all sorts of conversions – but not, as I fondly hoped, file format conversion, so that I could read MS-DOS disks; instead, all sorts of numeric conversions (Hex-Decimal, Decimal-Hex). It also gives control over loading and saving sections of memory. This Tools facility did seem to be unique to Qtop, and provides useful utilities for those engaged in the relevant activities.

Options

Options enables you to set various parameters and defaults in a well-planned and functional way. It is here, for example, that you designate the programs to go into the *My Progs* menu; here you can reset the Qdos clock, the network station number; and so on. You also save all your configuration information from here. Interestingly, you can also load different configuration files from here, so long as you first change the TD-Device: each configuration file must be called Tdesk Opt, and therefore you cannot have more than one in a single directory. You can load a translation table at this point, which will be of considerable use to those using foreign alphabets.

Next I added to my boot file

```
9 LRESPR win1_edt_xtras
```

and reset the system to see if there was any problem with running *The Editor*. There wasn't: it seemed to slot in effortlessly,

and executing win1_edt_bin brought up the familiar screen instantly.

Although Qtop has some nice features, and appears well able to accommodate the familiar QL programs multi-tasking together, to claim that it has twice the power of Qpac2 seems to me excessive. In some areas – channel control, the viewing of file info – it is actually not as powerful. I felt that some of the facilities provided were either rather specialised – the numerical conversion, for instance, or memory saving and loading – although they are certainly powerful; or they are rather peripheral (the demos, perhaps).

One of the things I particularly missed was the ability to set up buttons. Qtop does have a sort of button facility, but the buttons seem but a shadow of those offered by Qpac2. In principle, you can hit F8 and be given a button for the current top window. This is very quick and simple, and works well with Tdesk itself and its associated files, but is problematical for many other programs such as *Quill* or *The Editor*, since the F8 key is often already

“*I was able to go ahead and make buttons which would appear in a net row across the top of the screen*”

assigned in such programs. Trying to change the system parameters for Button startup unfortunately only produced the message Fatal Error -23906. In addition, making a Tdesk button appears to be but a temporary measure, lasting only until the next use of the full window, and then requiring you to remake the button with another stab at F8.

So my next move was to try Qtop together with Qpac2, which allows a different, more permanent type of button with far fewer problematical clashes. I therefore added a line to my boot file

```
75 LRESPR win1_qpac_qpac2
```

and reset the system again. That all seemed to work beautifully, and I was able to go ahead and make buttons which would appear in a nice neat row across the top of the screen whenever I pressed both mouse keys simultaneously. I could, for example, attach the Qpac Files module to a button, and bring up the full power of that facility at the touch of a mouse key. Calling up the button a second time would then allow me to generate another Files

window – an extremely useful capability if you need to compare, say, a winchester directory with its floppy backup. I could not know that this multiple use of a button was possible via Tdesk, but Qtop did not seem to clash with Qpac2 in this respect.

Trusty old QL

While I was doing the review, I tried Qtop on my trusty old QL as well. I had recently upgraded this machine with Miracle System's splendid Gold Card, giving it a severe breath of new life. I had also bought Miracle System's 40 MB hard disk about a couple of years ago. In the event, there was no trouble getting Qtop to work with the following (partial) boot file:

```
100 TK2_EXT
110 LRESPR win1_edt_xtras
120 LRESPR win1_cowo_thorql_bin:
REMark Required by Tdesk
130 LRESPR win1_qram_ptr_gen
140 LRESPR win1_qram_wman
150 LRESPR win1_cowo_hot_rext
160 LRESPR win1_atari_ataridos_rext
170 LRESPR win1_qtyp_spell
180 LRESPR win1_qpac_qpac2
190 LRESPR win1_tools_qload_bin
200 HOT_GO
210 setpicks: REMark Procedure- sets
hotkeys
220 mkbutton:REMark Procedure- sets
up buttons
230 EX win1_cowo_Tdesk; 'win1_cowo_'
240 WIN2 psion : A_BLANK
```

Unfortunately I don't have a mouse for my QL, and it is more fiddly driving Qtop with the cursor keys. Qpac2, on the other hand, has been designed very carefully with non-mouse users in mind, so that there is far less discrepancy between mouse-driven and cursor-driven operation. Still, Qtop did seem to work without trouble in this environment.

Duplicated

With Qpac2 in operation, much of the functionality of Qtop is duplicated, and I found myself asking whether I really needed it at all. It seemed to work swiftly, to be intuitively designed, particularly for mouse operation, and did not feel labyrinthine or over-complicated. But it is obviously aimed first and foremost at the THOR, so that for an Atari-user there are parts of the package which are redundant.

What authors put into an application no doubt reflects their overall preoccupations, so perhaps for Urs Konig Qtop provides twice the power of Qpac2, but my overall judgement – admittedly after less than a week – is that it adds little of significance to what I personally get out of Qpac2, and in some respects, it lags behind.

PUBLIC DOMAIN Games

I Rich Mellor spies on the games you can have for a song.

For any user new to a computer, the public domain scene can provide a source of cheap games programs which, although varying in quality, can be a quick way of providing entertainment for young and old alike, and form a useful introduction to the workings of a computer and allow you to get used to the keyboard.

The QL is no exception in that there are quite a lot of public domain games available which range from pretty awful to very good in quality. Although CGH Services has provided a wide selection of games software for quite some time, there is now a newcomer to the public domain scene, Qubbesoft, which lists a few games in its catalogue (some of which are the same as those in the CGH library).

Qubbesoft's library is split into several disks (or microdrives) that each contain a small boot program to access the supplied programs. Although this would not be much use for programming utilities, in the games area of the public domain market, this is a very good idea. It certainly makes the ordered disk/microdrive a lot easier to understand (and use) than the way in which CGH public domain programs are sorted (in batches). I must admit to liking the way in which the Qubbesoft library is set out and would suggest that CGH could learn from this where games and adventures are concerned.

There are several versions of *Breakout* available from both libraries. For those of you who were not au fait with the computer scene in the early 1980s, this was originally a very simple game where a wall of bricks appears at the top of the screen. You are given a bat which can only be moved left or right, and then it is your task to keep the ball provided in play and knock all of the bricks out of the wall.

Although a relatively simple game, all of the different versions supplied in the library are pretty good and should keep younger games players quiet for a while. However, I have yet to find a version to beat the Psion game which was given free with some QLs. The Psion program has the advantage that it was written mainly in pure machine code and is therefore compact, fast and varied (what a pity Psion have not consented to making its program public domain).

Some veteran ZX81 users may remember an old maze program called *3D Monster Maze* – an excellent program that pushed the ZX81's block graphics to their limit. On the QL there is even a version of this program genre in the public domain,

still prove a challenge to those new to the computer. However in these more hi-tech days, these sorts of programs only tend to appeal to the very young for a very short period. Programs available for the QL include *Blackjack*, *Space Invaders* (I still prefer the QL World version, *Paladin*), *Tron*, *Bomber*, *Othello* and *Star Trek*.

More exciting programs available from the Qubbesoft library include those games which were originally given away free by Medic in the early days of the QL. Unfortunately these programs begin to show the possible hazards in maintaining a public domain library. Although they were given away free, the ex-owner of Medic having been given a prison sentence and the author uncontactable, they

are probably not in the public domain as such since the author 'Janko Mrcic Fogel' (who actually wrote several good arcade games for the QL in the early days) most probably still possesses the intellectual property rights to the programs.

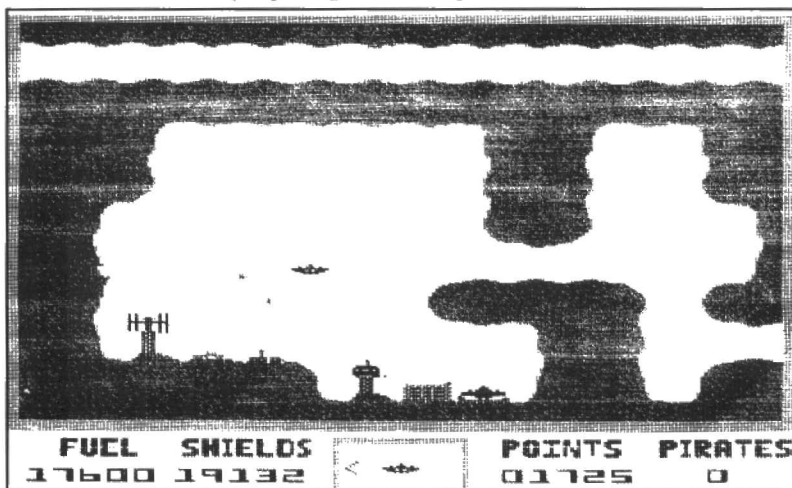
Anyway, on to the different games.

Treasure is a game made up of lots of different screens containing platforms. The idea is to collect all of the bags of gold from each screen. In your way are various moving

objects which will kill you if you touch them; lifts between platforms (together with trampolines), umbrellas and stairs. This game is not easy and is actually very good. Although it is a cheap way of introducing someone to this type of game, do not be too disappointed if you cannot finish this, because in common with the better known *QL-Cavern* this program cannot be finished either.

Medic's 'Pacman' clone is also available from Qubbesoft. This is a very fast version with lots of different mazes for you to clear the food pills from. There are four ghosts to avoid. This takes a lot of skill and speed to complete, but is highly recommended. The graphics are excellent and should keep many a player occupied for a long time.

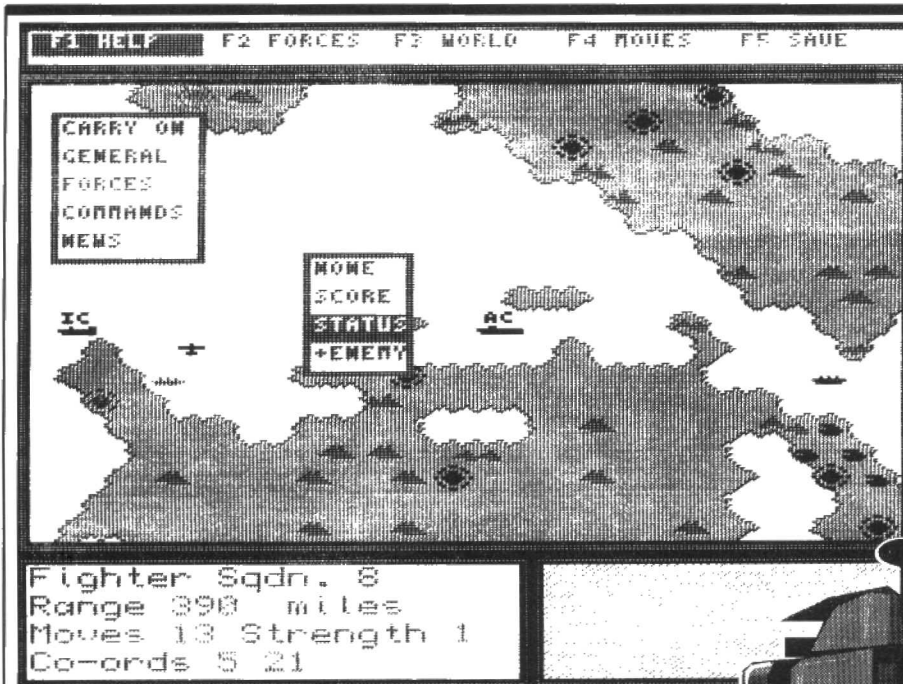
Returning to those games which are definitely in the public domain:



Starburst – pirates and guns in caverns.

Cyclops Maze. In this program, you are given a random maze which is shown in three dimensions on screen. Unfortunately, there is no monster chasing you around the maze, but it still proves quite a challenge as you search for the Cyclops' eye. The maze is not too testing and users may wish to try to alter the Basic program to create a larger maze. This program is available from both public domain libraries. However, for a more challenging 3-D maze puzzle, games players may prefer to get the version by Alan Pemberton (well-known for good adventures) which is in the CGH library (*Angstrom*). I have not actually seen this program, but knowing Alan as I do, I can only assume that it will be well worth the copying fee!

Indeed public domain libraries are full of renditions of these old favourites which



War - attacking the enemy

CGH Services lists in its library *Starburst* which was originally sold for £12.95. This is a very welcome addition to the public domain scene. It is in 100 per cent machine code, and consists of 256 caverns, around which you must fly your plane in search of pirate space ships. You have seemingly plenty of time to fly around and located the pirates, however, as time progresses, you soon find that you are flying around in circles as the maze of caverns becomes more intricate. This is a highly addictive game and recommended for those on a tight budget. I have managed to locate all of the pirates and even the final mother ship, but unfortunately not all in the same game (luckily the maze does not change and so I should be in with a better chance next time).

CGH also boast an early version of *Tetris* (originally written in French) in its library. This is quite a good version of this program, but I must admit, has a long way to go having played CGH Services commercial version of this game (*Double Block*). Nevertheless, the game on which this is based is compulsive and very good for your mind, therefore at least one version should be in everyone's library.

On the shareware side CGH have three excellent programs, which should cater for most of the QL community.

Cavern Frenzy is a 'Boulderdash' clone. This game made up of 15 different screens which are filled with boulders and jewels. The idea is to collect all the jewels on one screen to open the door to the next level. Unfortunately, the boulders are supported only by little blocks of earth. When you move this earth, it causes a minor landslide of the boulders which can all too easily kill you. After a bit of use, you begin to learn where the boulders will fall, and can therefore use this to your advantage. Luckily there is a training mode in which you can go through the door to the next

level without having collected all of the jewels on that screen. Even in the training mode, I have been unable to complete half of the 15 levels and so can only suggest that you try this one for yourself. A great game for kids.

Worm Germ is written by one of CGH's less mainstream programmers. This is a clone of the old centipede game where you control a centipede which must be guided around the screen to eat the food littered about. As you eat, your body grows in length. The idea is therefore not to get trapped and thereby bite yourself. This game is pretty fast and has many different features which should recommend it to both young and old.

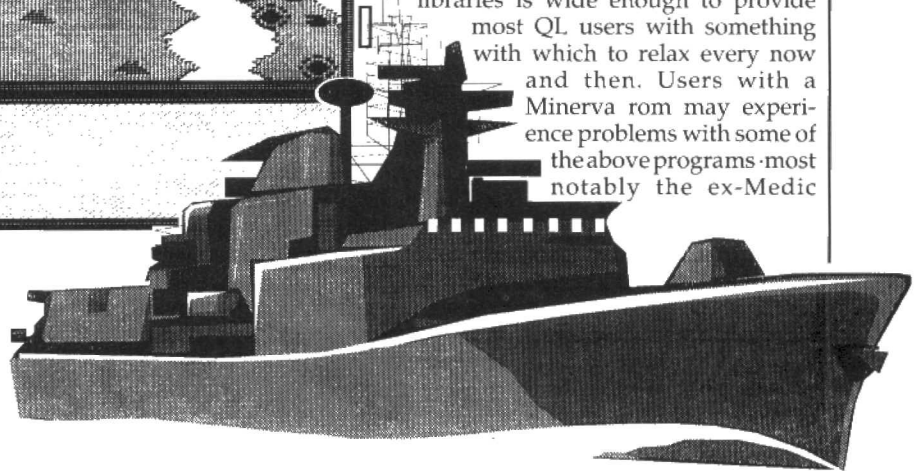
War-games

Last but not least, provided that you have Toolkit II, there is one war-game in CGH's shareware library. Unlike the public domain *Supremacy* which is similar to the 'Risk' boardgame, this program is more of a challenge. *QL War* consists of a Turbo-charged Basic program which is menu driven to allow you to send troops by ship or plane to far and distant lands hopefully to conquer new continents and keep the enemy hordes at bay. You can choose whether to play against the QL or even another player on another QL, and after choosing the size of the map on which you are to play, you are put in control of quite a sizeable army.

This program takes some getting used

to, especially due to the fact that enemy units remain hidden until one of your own units is within one square of them. However, this forms the basis of what could be a very good commercial program (if only the author had enough time and encouragement); and it will appeal to a wide range of war-gamers, who will not only have to control their army, navy and airforce, but also capture towns and put them to work to produce further armies. Unfortunately, not a game for the pacifist (except as a sounding-board for the futility of war), but in all one worth trying.

As you can see the variety of leisure software provided by public domain libraries is wide enough to provide most QL users with something with which to relax every now and then. Users with a Minerva rom may experience problems with some of the above programs - most notably the ex-Medic



games), although these programs are quite simple to fix. I have supplied details of the fixes to these programs to CGH Services for publications in *QL Leisure Review*.

There are still lots of other programs out there which could be made into public domain programs and enhance this scene even further. I have seen other public domain software which is not present in either of these libraries, but cannot think where I obtained it - perhaps the saddest loss here was a fully fledged *Defender* clone (a little too hard for me to play but very good nevertheless). If you are the author of any commercial programs which have not sold many copies for a while, or indeed are thinking of writing a piece of leisure software which you do not think will sell, but could interest someone else (especially in the field of education) then why not get in touch with a public domain library?

INFORMATION

QUBBESOFT PD LIBRARY

K Dunnett, 38 Brunwin Road, Rayne, Braintree, Essex CM7 5BU. Tel: 0376 47852.

Price: 30p per mdv/50p per disk (you supply media).

CGH Library

CGH Services, Cwm Gwen Hall, Pencader, Dyfed, Cymru SA39 9HA. Tel: 0559 384574.

Price: £1 per microdrive-sized batch.

DIY TOOLKIT



Simon Goodwin unravels his MORE extension and explores the QL's '32 bit' credentials.

The command MORE is a quick and easy way to look through any Qdos file, designed as a replacement for VIEW and COPY TO SCR. MORE displays pages or lines, showing the position of the data in the file and allowing fast navigation.

The machine code for MORE was presented in last month's *QL World*, along with the first part of the assembly code, concerned with initialisation and parameter checking. The accompanying listing unveils the display and input routines that show any page of a file under keyboard control.

MORE uses SuperBasic's buffer area to store details of the file and windows in use. The diagram reveals the way the space is allocated, using some parts to hold several items in succession.

The space used to assemble the file name and DATA_USE default holds the file header once the file is open; as soon as the prompt line has been written, text from the file re-uses the same space, leaving only the file length for later checks.

MORE includes concise display and input routines for 32 bit integer values. These are faster and more predictable than their floating-point equivalents, but they push the '32 bit' attribute of the QL instruction set to its limit, as we shall see.

MORE STUFF displays pages of text from STUFF, accepts key presses and writes prompts in the command window, #0. There are five control keys: DOWN ARROW, PAGE DOWN, PAGE UP, ENTER and ESC; these can easily be reconfigured by modifying the program. The code that handles each key is discussed later.

The speed with which MORE displays each page is impressive even on the most humble 128 K microdrive system. This is because of the way that MORE uses Qdos, reading and writing groups of lines with a single system-call. Other programs are slower because they perform several calls for each line.

The code to draw each page is short but

QL WORLD DIY TOOLKIT SEPTEMBER 1991 - Listing, page 1 of 4

```
* QL WORLD DIY TOOLKIT - MORE keyword, by Simon N Goodwin
* Continued from LISTING ONE of AUGUST 1991's DIY TOOLKIT
* Redraw the screen forwards from the file to the main window
*
redraw      exg      output_id,a0      Select the main output channel
            moveq    #32,d0            SD.CLEAR - clear output window
            trap     #3
            exg      a0,output_id      Re-select the input file ID
            move.w    depth(buf,a6.l),d4
            move.w    width(buf,a6.l),d6

next_line   *
            * See what is already in the buffer (D1 bytes at TEXTPTR)
            *
            move.w    textleft(buf,a6.l),d1
            beq.s     get_some
got_more    move.l    textptr(buf,a6.l),a1
            moveq     #10,d0           Preload code of a line end marker
            subq.w    #1,d1
            bmi.s     run_out          Get some more text
            cmp.b     0(a1,a6.l),d0    Check one character
            addq.l     #1,a1           Advance to next character
            beq.s     got_line         Line end found
            subq.w     #1,d6           Reduce remaining width
            bhi.s     scan_line

            *
            * The line is 'full', but we may be able to squeeze in a line end yet
            *
line_full   tst.w     d1               Is the next byte in the buffer?
            beq.s     got_line         Trust to luck (and fail safe!)
            cmp.b     0(a1,a6.l),d0    Is it a line end?
            bne.s     got_line         If not, leave it for later
            addq.l     #1,a1           We can accommodate that too
            subq.w     #1,d1
            width(buf,a6.l),d6        Recall line width for later
            move.w     #1,d4           Count one less display line
            subq.w     bhi.s           Continue if there's anything left
            move.l     a1,d2           Work out number of bytes scanned
            sub.l      textptr(buf,a6.l),d2
            sub.w      d2,textleft(buf,a6.l)
            bsr        send_bytes      Display scanned buffer bytes
            move.l     a1,textptr(buf,a6.l)
            bra.s      find_ptr

            *
run_out     move.w     textleft(buf,a6.l),d2      Display all we have
            bsr        send_bytes      Write buffer contents to display
            get_some    move.w     maxtext(buf,a6.l),d2
            move.l     textbuf(buf),a1          Reload the text buffer
            move.l     a1,textptr(buf,a6.l)
            moveq     #3,d0             IO.FSTRG - read D2 bytes to A1
            bsr        do_trap4
            fetched     beq.s      fetched        No error, OK!
            cmpi.b     #-10,d0           End of file?
            bne.s      file_error         Anything else should be reported!
            move.w     d1,textleft(buf,a6.l)
            beq.s      find_ptr           Nothing left to read?
            bra.s      got_more           Carry on with what's left
            file_error  move.l     d0,d2       Note the error code
            bsr        closer             Try to close the file (preserving D2)
            move.l     d2,d0             Return the file error to SuperBASIC
            rts

            *
            * Find file pointer and adapt to suit TEXTLEFT
            *
find_ptr    moveq     #0,d1             No move
            moveq     #67,d0           FS.POSRE
            trap      #3               Set D1.L to file pointer

            move.w     textleft(buf,a6.l),d2
            ext.l      d2
            sub.l      d2,d1           Don't count bytes not yet shown

            *
            * Seek start of line in report window and show file pointer value
            *
```


tightly wound. At the start of the listing the main variables in BUFFER are set, but the output window is untouched and no data has been read from the file. REDRAW starts by clearing the window and noting its width and height in registers D4 and D6. These are used to work out the amount of text that will fill the window.

The value at TEXTLEFT is zero, so the buffer is filled by the code labelled GET_SOME. This simply sets the registers expected by the Qdos input routine IO.FSTRG, and tries to read MAXTEXT bytes to the buffer at offset TEXTPTR inside SuperBasic.

If the call to DO_TRAP\$ goes smoothly TEXTLEFT can be updated with D1, the number of bytes read, and processing continues at GOT_MORE or FIND_PTR as appropriate; otherwise MORE tidies its channels and reports the file error.

The heart of MORE is a nested loop that scans the input, counting characters for each line and starting a new line as appropriate. The code takes pains to record 'what's left', whether that be space on the line, lines on the page, bytes scanned or unchecked in the buffer or in the file.

If the whole buffer is scanned and will not fill the window, the code at RUN_OUT displays the text left, before reloading the buffer with the familiar GET_SOME routine. Often the data is ready and waiting in QL slave blocks, thanks to 'pre-fetch' routines in directory device drivers.

Normally the Enter character marks the end of each line, but MORE works with _DOC, _DBF and code files that may lack such signposts. Qdos also moves to the next line when a line is filled with characters, but avoids making a blank line if the next character is Enter.

The code at LINE_FULL looks onward in the buffer to see if it can squeeze an Enter onto the end of a line. This check is problematic if the buffer empties just as the line becomes full, so MORE 'fails safe' in this rare case; it may generate a blank line, but it will not scroll the windows unexpectedly.

When the page is complete the code labelled FIND_PTR works out the position in the file from the file pointer and the number of bytes waiting in the buffer. MORE writes the current position at the start of the prompt line.

Successive digits blur together if overprinting is selected in that window, so you should favour the default, OVER#0,0 rather than OVER#0,1 or OVER#0,-1. This is little hardship; the alternatives are rarely used as they make it tricky to edit commands.

The lines labelled KEYROLL read and act upon the control keys. Errors or the code for ESC divert execution to STOP_NOW, closing the input file and moving to a new line in window #0.

The other control keys let you move through the file. They use the current file position, in register D4, comparing it with LENGTH, the file size in bytes from the file header. If you try to move past the end of the file execution is diverted to SEEK_START, winding back to the beginning of the file; otherwise the down arrow

```
*
update      move.l    d1,d4          Save file pointer for display later
            exg       prompt_id,a0  Retrieve prompt ID, save file ID
            bsr       line_start    SD.POS works like AT line,column
            move.l    d4,d1
            bsr       print_long    Print file pointer value

*
* Wait for ESC, Arrows, ENTER or ERR.NC and react accordingly
*
keypoll     moveq     #1,d0          IO.FBYTE - get 1 key!
            trap      #3
            tst.l     d0             BREAK? EOF?? #0 gone?
            bmi.s     stop_now       Errors here signal the end
            cmp.b     #27,d1         ESC?
            bne.s     dont_stop      Escape also stops the action

*
* Close input file and move #0 to a new line, ready for the next command
*
stop_now    exg.l     a0,prompt_id   Save prompt ID, select file ID
closer      moveq     #2,d0          IO.CLOSE
            trap      #2             No error expected
            move.l    prompt_id,a0   Now tidy up the prompt line
            moveq     #15,d0         SD.CURS - suppress cursor
            trap      #3
            moveq     #10,d1         Send Newline to #0
            bra       print_byte     Exit via Qdos TRAP #3

*
dont_stop   exg       a0,prompt_id   Save prompt ID, restore file ID
            cmp.b     #216,d1        Line Down (down arrow) ?
            bne.s     try_paging
            cmp.l     length(buf,a6.1),d4
            beq.s     seek_start
            moveq     #1,d4          Depth is one line
            bra       next_line      Scroll one extra line
try_paging  cmp.b     #217,d1        Page Down (ALT down arrow) ?
            bne.s     not_down
            move.l    top_left(buf,a6.1),old_top(buf,a6.1)
            move.l    d4,top_left(buf,a6.1)
            cmp.l     length(buf,a6.1),d4
            bne       redraw         Page forward
seek_start  moveq     #0,d1          Seek start of file
seek_d1     move.l    d1,top_left(buf,a6.1)
            moveq     #66,d0         FS.POSAB
            trap      #3

blank_out   exg       prompt_id,a0   Retrieve prompt ID, save file ID
blank_outp  bsr       tidy_wind      Blank the previous pointer value
            bra       none_left

*
* Page back; check TOP_LEFT; if at start of file, wind to near the end
* Move file pointer back to previous page (if known) or by set distance
*
not_down    cmp.b     #209,d1        Page Up (ALT up arrow) ?
            bne.s     not_up
            tst.l     top_left(buf,a6.1)
            bne.s     wind_back      Move back unless previously at top
            move.l    length(buf,a6.1),d1
            sub.l     #512,d1        D1 is 0.5K before the end of the file
            bpl.s     seek_d1        Wind the file pointer there
try_again   exg       prompt_id,a0   Restore prompt ID
            bra.s     keypoll        Eek! Await further instructions

*

wind_back   move.l    old_top(buf,a6.1),d1
            bpl.s     seek_d1        Use old top if it is valid
            sub.l     #2048,d4       Try to move back 2K bytes
            bmi.s     seek_start     Too far? Go to start instead
            move.l    d4,d1
            bra.s     seek_d1        Move backwards

*
not_up      cmp.b     #10,d1         ENTER key code?
            bne.s     try_again

*
* ENTER = Set new position
*
            exg       prompt_id,a0   Retrieve prompt ID, save file ID
            bsr.s     tidy_wind      Scrub the existing value
            bsr       line_start     Get ready to read a new value
            moveq     #10,d2         Buffer size
            lea.l     digits(buf),a1 Point at the buffer
            moveq     #2,d0          IO.FLINE, read up to D2 bytes
            do_trap4
            bne.s     bad_entry      Error, redraw!
            subq.w     #1,d1         Several bytes, please
            bad_entry ignore a null entry
            lea.l     digits(buf),a1 Point at the buffer
            bsr       long_value     Evaluate digit string into D1.L
            beq.s     move_there      If that worked D1 is the new position
            move.l    top_left(buf,a6.1),d1 Redraw last full page
bad_entry    moveq     #1,d3          Restore corrupted timeout
move_there  move.l    d1,d4
            moveq     #20,d0         SD.NCOL, cancel pending new line
            trap      #3
            moveq     #14,d0         Re-enable cursor
            trap      #3
            exg       a0,prompt_id   Retrieve file ID
            move.l    d4,d1          and required position
            bra.s     seek_d1        Move there
```



```

* D1.L to ASCII TEXT OUTPUT SUBROUTINE; A0 is Channel; Uses A1, A3, D0-D2
*
print_long lea.l    digits(buf),a3
           tst.l    d1
           bne.s    not_zero      Zero is a simple case
           moveq    #'0',d1      Digit zero
print_byte moveq    #5,d0        IO.SBYTE
           bra.s    call_qdos
not_zero  moveq    #9,d2        DBRA counter for up to 10 digits
op_digits moveq    #0,d0        Now buffer D1 as a 32
           swap     d1           bit unsigned integer,
           beq.s    enough
           move.w   d1,d0        last digit 'first',
           divu     #10,d0       from 9(A3,A6) backwards
           swap     d0,d1
           swap     d1          Divide the other 16 bits
           divu     #10,d1       Long integer division
           move.w   d1,d0       needs two steps pre-68020!
           swap     d1
           exg      d1,d0
           addi.b   #'0',d0      Convert digit number to ASCII
           move.b   d0,(a3,a6.l) Store at end of digit buffer
           subq.l   #1,a3        Move backwards through buffer
           dbra     d2,op_digits Count 9 to -1 for digits 10 to 1
enough    lea.l    neg.w        digits+1(buf,d2.w),a1  A1 -> first significant digit
           neg.w    d2
           add.w    #9,d2        Number length is 9 - D2 from DBRA
           trap     #4           Buffer is relative to A6
           bra.s    print_str

*
tidy_wind bsr.s      line_start
           lea.l    blanks,a1
print_abs move.w     (a1)+,d2      Pick up string length
print_str moveq      #7,d0        IO.SSTRG
           trap     #3           Display the text; no errors expected
           rts

*
windowspec moveq    #11,d0      SD.CHENQ trap key, window enquiry
do_trap4   trap     #4          Addresses are A6 offsets
call_qdos  trap     #3          General-purpose ROM caller
           tst.l    d0          Set Z to signal ERR.OK
           rts

*
line_start move.w    line(buf,a6.l),d2
           move.w    column(buf,a6.l),d1
           moveq     #16,d0      SD.POS
           bra.s     call_qdos    Move to character position

*
send_bytes exg      output_id,a0 Select screen output channel
           move.l    textptr(buf,a6.l),a1
           bsr.s     print_rel
           exg      a0,output_id Restore file ID
           rts

*
* ASCII TEXT to LONG WORD COERCION SUBROUTINE
* Text @ (A1,A6), D1.W = length, 1-9; uses D2 & D3; A1 points beyond last
* character on return, D1 is a LONG result and D0 is error code, 0 or -17.
*
long_value move.w    d1,d2      D2 is length un-scanned
           moveq     #0,d1      The result accumulates here
           moveq     #0,d3      LONG store for each digit
next_digit move.b     0(a1,a6.l),d3
           addq.l    #1,a1
           cmp.b     #'9',d3
           bhi.s     bad_value   Not more than '9'
           sub.b     #'0',d3     Convert ASCII to value
           bcs.s     bad_value
           add.l     d1,d1        D1:=D1 * 2
           move.l    d1,d0        Save D1 * 2
           asl.l     #2,d1        D1:= (D1 * 2) * 4
           add.l     d0,d1        D1:= initial D1 * 10
           add.l     d3,d1        Add digit to running total
           subq.w    #1,d2        One less character to read
           bne.s     next_digit
got_along  moveq     #0,d0      No error, result in D1
           rts

*
bad_value  moveq     #-17,d0     Error in expression
           rts

*
blanks     dc.w      10
           dc.b      ' '
           ' '           One space for each digit
spacer     dc.w      4
           dc.b      ' of '
           ' '           Spaces either side
lbracket   dc.w      2
           dc.b      ' ('
           ' '           One leading space
rbracket   dc.w      1
           dc.b      ')'
           ' '           No spaces
spacer2    dc.w      4
           dc.b      ' in '
           ' '           One space each side

*
define     dc.w      1
           dc.w      more-*
           dc.b      4,'MORE'
           dc.w      0,0,0      End of Procedure list, no FNs
           end

```

controls move you on to the next line or page, using residual data from the buffer and reading more as required.

When you page forwards MORE keeps track of the previous page offset, so that you can page back if you forget what you have just read. The top of each new page is recorded in TOP_LEFT, while OLD_TOP holds the previous value.

Backward moves are handled at NOT_DOWN. This starts by checking the current TOP_LEFT. If we are on the first page, TOP_LEFT holds zero and the 'previous' page is considered to be the end of the file. MORE winds to 512 bytes before the end - or any other offset you specify by patching the code - and redraws from there.

If OLD_TOP is set, WIND_BACK uses the value to find the new page; otherwise it subtracts an arbitrary step - 2048, by default - from the file position, and redisplay from that point. Notice that SEEK_D1 blanks the start of the prompt line with a call to TIDY_WIND. This ensures that backward moves do not mess up the display when the new position has fewer digits than its predecessor.

The last control key is ENTER, recognised at NOT_UP. This reads a line of up to ten characters into the DIGITS area of the SuperBasic buffer, and attempts to convert them into a position by calling LONG_VALUE. If any error occurs the screen is redrawn from TOP_LEFT; otherwise the number entered becomes the new file pointer.

Two TRAP #3 calls are used to tidy up after IO.FLINE; the first, SD.NCOL, moves the cursor but it is called to clear the window's 'pending newline' flag. Normal output after IO.FLINE takes a new line, but that would not be desirable in this case. SD.NCOL clears the flag, like all cursor positioning calls. SD.CURE turns the cursor back on, ready for the next command.

I often caution programmers about the need to check the error codes returned by Qdos. Programs may get stuck or corrupt memory if they continue without corrective action after an error. Yet careful study of the listing reveals several points where there is no check on the value of D0 after a TRAP instruction. This deserves explanation.

In general, MORE does not check for errors after operations that use the display channels. This saves a fair amount of code and should not be problematic as the first operations performed on each window are checked; thereafter the type and state of the channels are known, so further checks are superfluous.

If anything important goes wrong it is vital that the user should be warned by a message, and recover control. The keyword must not get stuck in a loop! For this reason MORE stops, with an appropriate report, if any error occurs when it tries to read a byte from the keyboard with IO.FBYTE.

The SuperBasic task generates a 'not complete' error if you press Break or Ctrl-Space while any of its channels are busy. Other reports are unlikely - but the unlikely is not the impossible.

File errors are trapped after the call to IO.FSTRG - any report except 'end of file' or successful completion terminates the

command. This check, between GET_SOME and FETCHED, detects unpredictable faults like memory exhaustion, disk 'read/write errors' or the dreaded 'Bad or changed medium'.

The last two subroutines in the listing are a matched pair. PRINT_LONG converts a 32 bit unsigned value into an Ascii decimal number between 0 and 4,294,967,295 ($2^{32}-1$), while LONG_VALUE packs a string of digits into the 32 bit data register D1.

These routines are interesting as they fill gaps in Qdos. Standard rom routines can evaluate long integer parameters, as well as the usual data types of string, floating-point and two-byte integers, but they cannot display or accept 32 bit values directly.

The QL floating-point format includes a 32 bit mantissa which will fit a long integer, but it is a waste to use clumsy floating-point subroutines when in essence we want to manipulate a single-processor register. The resultant code is needlessly slow and may give inappropriate results.

The interpreted SuperBasic command PRINT 12345678 uses a complicated floating point calculation loop to normalise the value, eventually returning 1.234568E7 rather than the text you might expect. Megabyte files are rare, and the time spent displaying numbers is a small part of the effort expended by MORE, but these long integer routines are good examples, potentially useful in other programs.

Like the popular 68000, the QL 68008 processor has a 16/32 bit arithmetic and logic unit. It can perform some 32 bit instructions, but the most intricate ones work on a mixture of 16

and 32 bit values. On this basis the Z80 and 8080 were 8/16 bit chips, and the 6502 may be the only true 8 bit processor.

It takes a 68020 or later model to perform 32 bit arithmetic in one step. QL code tricks are needed to multiply and divide 32 bits while processing each digit of a long integer value.

PRINT_LONG works by repeatedly dividing its parameter by ten, generating a new digit from the remainder after each division. Digits are stored in reverse order in the buffer, so that the least significant digit is the first to be generated, and stored at the end. The value zero is treated as a special case, output directly with IO.SBYTE; otherwise the code at OP_DIGITS loops till there is nothing left to be divided.

This long division is complicated by the need to handle values greater than 655,350. Motorola's 68008 divu instruction takes a 32 bit value and divides it by a 16 bit quantity; ten, in this case. The result is two 16 bit values: the quotient or whole part of the result ends up in the low word, with the remainder in the most significant 16 bits of the register.

This works fine unless the initial value exceeds 655,350. Then the result is too big to fit in the low word, and an overflow error occurs. The conventional solution is to perform the division in two steps, manipulating half of the 32 bit value each time. You could trade space for speed by checking the value and using a single divu in a loop for small numbers.

A similar problem is avoided by LONG_VALUE, which packs text into 32 bits. The result accumulates in register D3;

each digit adds to the running total, which is multiplied by ten to make room for the next.

The 68008 mulu instruction takes two 16 bit values and generates a 32 bit result. Rather than use mulu #10 twice, and shuffle register halves as in PRINT_LONG, I use a technique learnt at the keys of early microprocessors that lacked multiplication hardware.

The 68008 can shift, add and subtract 32 bit values, although it cannot multiply them in one step. My code multiplies D1 by ten by adding it to itself and storing the temporary result, then shifting that twice to the left, in binary. This leaves two values ($D1*2$ and $D1*8$) which equal $D1*10$ when added together.

This is shorter and faster than a pair of mulu instructions and associated SWAPs; I also believe that it is easier to read and verify. MORE limits numeric input to ten characters, including the ENTER, so there is no need to check for numeric overflow in this application. Add BVS.S BAD_VALUE instructions after each ADD or ASL if you re-use my code and need to detect overflows.

If you like the sound of MORE but do not fancy typing the listings into your QL, ST, Amiga or Thor you may be interested in **DIY Toolkit Volume V** for file Viewing. This software set includes the code, documentation and complete source for MORE, plus Luca Pivato's MEM device.

Volume V costs £7 on floppy disk (please state size) or your cartridge, from DIY Toolkit, Cwm Gwen Hall, Pencader, Dyfed, Cymru SA39 9HA, tel: 0559 384574.

KEN BAKER SOFTWARE

32 Hunts Pond Road – Park Gate – Southampton – SO3 6QA – 0489 581056

All Programs for 768K Expanded QL with Trumpcard

SPEEDSORT

The ultimate sorter! Sorts arrays of any size by given element number. The speed is devastating – up to 80,000 sorts per second!

Machine Code £20.00

QLDL

QL Disk Library allows the user to catalogue up to 3000 program titles (around 75 disks at a time) in alphabetical order and cross-referenced by numbers given to the disks. No more searching for endless hours for that lost program.

Machine Code £15.00

SCREENMASTER

Screenmaster will record anything currently on screen in a 'squeezed' form, and reproduce the picture in a variety of cinematographic-style wipes and speeds.

Machine Code £20.00

PLANNER

One of the most useful programs you could wish for. It will provide a printout of dates, with day and week numbers, in a month-per-page format from a given starting date and number of forward days. It also includes routines for finding day and week numbers you can use in your own programs.

Machine Code £10.00

TRANQUIL

Tranquil converts Quill documents into BASIC PRINT statement form in order that they can be more easily manipulated for printing and adding printer instructions etc.

Machine Code £10.00

ALL PROGRAMS SUPPLIED ON TOP-QUALITY 3 1/2 INCH DISKS – POSTAGE & PACKING FREE

QUESTIONNAIRE THE STORY SO FAR

As the questionnaires roll in I thought that it might be useful to review some of the impressions gained so far as well as provide some background. Out of perhaps some 10,000 readers I must first say that the response, until now, has been poor. This has important repercussions when it comes to analysing and generalising from the questionnaire, for if only 200 reply, then it is possible that the sample may be highly unrepresentative. If, for example, only people with two QLs and a disk drive reply, then the erroneous impression is given that this is a very common configuration when in fact there might only be 400 disk-drive owning QL users altogether. If the same personality factor that persuaded them to buy a disk drive and a second QL also persuades them to send in a completed questionnaire then they become over-represented. This *sampling bias* is something that is inherent to this type of questionnaire and hangs over any interpretation made. The only way to tackle this would be if a significant proportion of QL owners/QL World readers actually sent in their questionnaires. By significant I mean 60%. It is up to you – and it is still not too late.

Turning first to the impressions gained of the respondents – for the reasons outlined above, not QL owners in general – from a quick and non-scientific examination, I must say that I have been pleasantly surprised to discover that there appear to be a large number of professional users still active with the QL. Taking into consideration their incomes, it would appear that many individuals have stuck by the QL despite the presence of the PC-compatibles. For example, there are General Practitioners (family doctors) who continue to use the QL to run their surgeries using an Archive-based database. This is despite the fact that there are a number of companies who will provide pc-based systems at thousands of pounds cost, and despite the fact that many GPs have been persuaded to part with this kind of money for the promise of sometimes very inadequate 'support'. However, it was pointed out that the speed of the QL was becoming more and more of a problem and that if the Miracle Gold card had not

I Dr. Sohail Bhatti reports on the replies and says: good for the two percent. Where are the rest of you

appeared they would have been forced to move over to the PC.

Other professionals that are prominent are teachers ranging from primary school to university professors, managers who judging by their incomes must be at least middle-ranking in large companies and also those who regard themselves as listed under 'computing'. Unfortunately, a large proportion of respondents omitted to write down their actual occupation description in addition to their job category. It must also be said that there is a significant proportion of respondents who are retired, so that many users are over 65. My impression is that most users are over 25 with the youngest not really being interested in a 'SuperQL' – being more attracted by the delights of the Archimedes or Amiga 3000. However, the youngest also seem to have the most hardware and/or machine-code skills.

I have also been pleasantly surprised to discover that most users have incomes above the national average and this is borne out by the fact that there are very few unemployed individuals who responded. Again, this might be because these people felt so ashamed at admitting this that they have not sent in their questionnaires, but I think that the sample is sufficiently large so that this would be inherently unlikely. Many people did point out that the boxes in the last column of 'Occupation' were missing. This was an oversight when typesetting. It is also worthwhile noting that there was no category specifically for the unemployed.

Most respondents had been with the QL for over two years and the greater portion of these fell into four years plus. This was reflected in their commitment to the QL as a substantial minority had more than one QL, dressed in different versions of the rom. One enthusiast even has seven of the machines! What he chose to do with them all will have to await further analysis.

Another surprising aspect of the respondents is that a large percentage appear to have 3.5in disk drives. I suspect that this reflects more on selection bias than an actual fact. Overall, there appear to be an impressive number of different peripherals plugged into the QL, with the Tandata modem and Miracle Trump Card being prominent among them.

As the questionnaire was pre-circulated to most Quanta sub-groups I was not surprised to find that they used the QL World version to reply. This meant that about a third of respondents appear to know other local users. In view of the geographical spread of users most seemed to know a small number, rather than large groups. Indeed, in general, it seemed that if you knew more local users you had a greater commitment to your machine in the way of peripherals and programs. In a way, this is rather unsurprising but does highlight the attraction of creating local QL groups. An obvious way to identify a local group is to join Quanta or failing that ask them where your local sub-group is held and who organises it. According to my information you will not be excluded if you are not a Quanta member, but having been a member for a number of years I find its members have been very supportive especially in the context of local sub-groups.

Moving on, there were a few people who were using Ataris or Amigas but almost all the respondents did not have a machine other than the QL, running QL-specific programs. In contrast, a significant proportion did admit to using pcs. These came in various hues ranging from the humble Amstrad PC1512 to 386-based goliaths. Judging by the hours spent on these machines, most seemed to use them at work, preferring to use the QL at home. While detailed cross-correlation will have to await full analysis, I did feel that most users of pc emulators seemed to fall into this category. Surprisingly there were a few individuals who claimed to be proficient in MS-DOS but who appeared never to use any pc-compatible machine!

Since this is nominally the summer, the number of hours spent hunched up to the QL did appear to be around two to four hours on average. A number of respond-

ents did point out that this would be more in winter, presumably when the rain is colder. The range, though, was massive with some individuals spending more than five hours a day on the QL. In view of the fact that only active users would be motivated to fill in the questionnaire I was not surprised to discover that hardly anyone claimed to use the QL for less than an hour a week. When it comes to using the QL, games appeared to be low on the list. The largest group of respondents gave wordprocessing the highest marks and most of these continue to use *Quill*. Again, it was frustrating not to know whether they meant ordinary *Quill*, *TurboQuill* or *TurboQuill Plus*. Surprisingly, a large group of respondents appear to be engaged in program development – and this seems to be in SuperBasic, judging by how they had graded their expertise in various fields. Many, though, appeared not to have learnt the benefits of the SuperBasic compilers or perhaps used Supercharge which was not listed as an option. This concentration on SuperBasic was emphasised by the number of people wishing more on the subject in QL World. However, there were also significant numbers wanting more on the principles of machine-code and Qdos.

I had thought that there were more people who had significant software portfolios but it seems that most people have only spent between £1 to £500 on software. This works out at less than £100 per year per user. Most appeared to have *Toolkit 2* but there was otherwise a wide spread of

program ownership. Unfortunately, since there were so many different titles it was only possible to select a few and this might itself reflect personal preferences on my own part – though I did attempt to cover a range of companies and products. Detailed analysis of this will have to wait until a future article.

Question 13 was an attempt to ascertain what the majority of people wished for most and it seemed that the Gold Card was high on almost everyone's priority as was the need for better screen resolution (which I believe Miracle are currently working on) and a hard disk. A few did point out that the cost of the hard disk and Gold Card would mean that they had forked out as much as an Apple Mac would cost.

Of course this would not come with a hard disk but the point was well made. In defence of Miracle, I must say that the extra-high density drives (now available as a bare mechanism for around £100) are as fast as the Miracle hard disk and have the added advantage that you can have three on one Gold Card and that the media, now £8 per disk is also removable. This comes out to £180 for 32 Mbytes – comparing favourably with standard pc prices.

The rest of the questionnaire was much more unstructured and so only anecdotal findings can easily be discovered. For example, quite a few respondents claimed to have difficulty obtaining a copy of QL World from newsagent shelves. It was

thus not unexpected that most respondents obtained their copy through subscription. Most also thought that QL World was well balanced though there were people with their own specific axes to grind. Quite a few did point out in their responses that they had little else to compare QL World to and that they had awarded high points by default. Hardly anyone had heard of *QL Technical Review*, but Quanta appeared to have a higher profile.

I hope that this article has whetted your appetite. When the analysis and tabulation of the QL World questionnaire is complete I certainly intend to share with you other interesting findings. It might be worth pointing out that the questionnaire has now been published in various forms in the UK and also abroad and there will be interesting comparisons to be made both within and between these various groups. Since there does not appear to be an appropriate forum to air these findings the overall analysis from all questionnaires that were circulated will be made available on a named basis, particularly to those who are members of Qlaw.

In any case, I can only urge you to help make the questionnaire as successful and as generalisable as possible by completing and sending it in. It is still not too late to pull it out of the July issue. I can also strongly reassure you that your name and address will never be shared with any extraneous company without your permission.

NEWSAGENT ORDER FORM

By placing a regular order with your newsagent you can be sure of receiving **Sinclair QL World** every month. Complete this coupon and pass it on to your newsagent.

Dear NEWSAGENT,

Please reserve/home deliver _____ copies of **Sinclair QL World** every month for:

Mr/Mrs/Miss

Address

Telephone No

Sinclair QL World is published on the 3rd Thursday of every month.
Any distribution queries to: **IPC Marketforce, Kings Reach Tower, Stamford Street, London. SE1 9LS. Tel: 071 261 5000.**

ARCH ED PART ONE

Stephen Mitchell presents an all-purpose editor that runs within Archive

If you have created any *Archive* database files, either using your own procedures or by running on application, you may have found it difficult to make changes subsequently to data in those files. This may be because the processes haven't catered for certain eventualities, or that some permissible changes are awkward to effect. In either case, an all-purpose editor that could run within *Archive* would make these changes less troublesome. This is the first of two articles that will provide you with such an Editor.

Thing of the past

Resorting to manual *Archive* Commands like 'alter' or 'update' to make ad hoc changes will also be a thing of the past. Indeed, the 'alter' command is not designed for use on records that have made use of 'long' fields. Data overwrite occurs on such fields on the screen and you are in

danger of corrupting your data. This Editor also overcomes this problem.

Greater use of 'long' fields became possible with the issue of *Archive* Version 2.00, which cured some of the problems that had existed in *Archive* Mk1 versions.

Text fields (string variables) within dbf files containing up to 255 characters each can now be created and handled without too many problems and without much fear of subsequent loss of any carefully scripted text. Long field creation and manipulation can only be reasonably managed from within *Archive* procedures, since no textual editing facilities are available directly in *Archive*'s particular brand of Basic. Indeed, within *sedit* screen layouts in Version 2.00, there was still an imposed restriction on the direct use of string variables.

The useful 255 characters length of such variables was chopped only to screen width (80 or 64 characters depending on mode). In this case *sedit* variables had to be concatenated if long fields within dbf

files were to be used. This has now been overcome with the issue of *Archive* 2.36 and *sedit* fields of the full 255 characters can now be handled.

However, it is still true to say that *Archive* lacks a generalised text handling facility. If a dbf record is designed to take advantage of this greater available space it is still necessary to design a number of complementary *sedit* screens in your application so as to display and manipulate textual fields whose joint size exceeds one screen (80x25 characters: approximately 7 fields, each of 255 characters).

All as long fields

The set of *Archive* procedures presented in these two articles are known as 'Archive Editor', or 'ArchEd' for short. They enable all text fields of a file to be treated as long fields and to be manipulated with comparative ease. I wouldn't claim that the standard of the facilities offered by the

```
ARCHIVE EDITOR                                     Memory 22758 Bytes
field1$:      This test file can help you practise using ARCHIVE EDITOR
               (ARCHED) before using it to update your own 'hard-crafted'
               database files. I hope you find ArchEd useful and that it
               enlivens your interest in using Archive .....
               Stephen Mitchell.
               i P.S. You might like to practise deleting /
replacing / inserting some of the 'words' in the next field.
field2$:      0 1 22 333 4444 55555 666666 7777777 88888888 999999999
               888888888 77777777 666666 55555 4444 333 22 1 0 a bb ccc dddd
               eeeee fffffff ggggggg hhhhhhhh llllllll hhhhhhhh gggggggg ffffff
               eeeee dddd ccc bb a

ENTER to edit text(i/d/r/dl=insert/delete/replace/del&ins);
SPACE ENTER at next reference point to re-edit current field;
SPACE SPACE ENTER " " " " " to accept field edit;
1,2,...,9,(1)0 to accept field edit & skip 'n' fields (cyclic);
F4/F5 to abandon/commit all current field edits & Exit.
```

procedures come anywhere near those expected of a word-processor, but some word-handling characteristics are present. Using these procedures, you no longer need to design and set-up specific Archive screen layouts, nor write the handling procedures to edit a file. Such screens and procedures would, no doubt, be peculiar to the structure of your file and need redesigning/re-writing to handle different files. In contrast, the procedures supplied only require minor alteration so as to reflect the names of the fields within your file and can be adapted for any file that contains textual fields that you may wish to edit.

The procedures contain no facilities for editing numeric fields.

Numeric fields contained within your files are ignored by the procedures and remain unaltered. To recap, ArchEd is a universal editor of text based Archive files.

In making these procedures available I make no apology for their lack of speed nor for any other detrimental feature attributable to the Archive environment. I would say however that, during development, I have endeavoured to strike a balance between speed, features and retention of memory available to Archive. The procedures have been developed within the confines of an unexpanded QL so you should have no problem running them under Archive. Having said that, I have found the restrictions somewhat severe. If you have an expanded QL then you might like to try merging all the procedures and running them as a single loadable program. ArchEd runs more efficiently in this form and it is how I use ArchEd to edit dbf files on my own QL.

Mainframe-type editor

The procedures together form an unsophisticated Archive 'editor' for textual fields. There is nothing new in the form of this editor. Readers who have some experience of mainframe and mini computers will recognise the approach taken in the type and use of commands in this editor. The editor is basically a 'line' editor with editing facilities available in forward movement through the fields within a record. In either case it is possible to cycle back to re-edit fields (or to edit fields that were previously unaltered) at either the beginning or the end of the record. The editor lacks many of the sophistications associated with both 'line' and 'screen' editors. There are no repetitive 'find and replace' type functions.

As a cautionary note, don't expect anything in the same league as the *Metacomco Editor*, *Quill* or *The Editor*. However, if desired, ArchEd can be incorporated within your existing Archive procedures to give greater flexibility to your database application.

```
proc start
  REM Save this listing as 'ArchEd_prg'.
  REM "ARCHIVE EDITOR - 'ARCHED' Version 2.02 3/5/91"
  REM Copyright S.G.Mitchell 1990
  REM
  REM Set your default filename here:-
  let file$="TESTFILE"
  REM Set the mode (mod=6 or 8) in which you wish to work here:-
  let mod=8
  mode 0,mod
  ink 4
  let c$="": let col=14
  if mod=6
    let wth=64: let fl=48
  else
    let wth=80: let fl=64
  endif
  wiz2head
  print at 3,8:"Version 2.02 (3/5/91)"
  print at 5,8:"Loading..."
  print
  run "ArchEd1"
endproc
proc wiz2head
  print at 0,0: paper 2:" A R C H I V E   E D I T O R ";
  print rept(" ",mod*3); "Memory ";memory(); " Bytes"
  if memory()<1200
    print at 1,mod*3:"**WARNING** Memory low - Close & backup file!"
  endif
endproc
```

```
proc start
  REM Save this listing as 'ArchEd1_prg'.
  cls
  wiz2head
  wib1esf
  run "ArchEd2"
endproc
proc wib1esf
  print
  wiz1yn:"Do you want to start a new database file?"
  print :wib2fnm: print
  if y:wib3crf
  wiz1yn:"Do you want to edit the file?"
  else : let y=1: endif
  if y
  wiz1yn:"Have you backed-up the file?"
  if y
  print : print paper 2:" WRITE privilege to file. "
  wib4opens
  return
  else : print "Back-up '";file$;"_dbf' before continuing."
  endif
  endif
  print "Press any key to return to ARCHIVE..."
  let a$=getkey(): mode 1,mod: stop
endproc
proc wib2fnm
  print "Enter the name of the file required as:-"
  print "  filename[ENTER] logical_filename[ENTER] (No quotes)"
  print "  [ The default (ENTER ENTER) is : fn = ";file$;" lfn = W"
  print "  resulting in :- Open '";file$;" logical 'W' ]."
  input "  filename = ";fn$: lfn = ";lfn$
  if fn$<>"": let file$=fn$: endif
  if lfn$="": let lfn$="W": endif
endproc
proc wib3crf
  create file$ logical lfn$
  refno
  field1$
  field2$
  field3$
  field4$
endcreate
let refno=0
let field1$="FIELD1"
let field2$="FIELD2"
let field3$="FIELD3"
let field4$="FIELD4"
-----
```


The main features of Archive Editor include:-

1. The full text of both the field being edited and the next text field in the current record is displayed. Some confirmation of the content and the style of the script being edited is therefore possible. In this respect the editor retains the main characteristic of a screen editor.

2. The current field may be re-displayed in its edited form before the edit to the field is accepted.

3. A record can be updated (committed) at any time between field edits. It is also possible to abandon all current field edits to the current record at any time. In practice it has not proved necessary to retain a feature included in an earlier version of ArchEd, namely to abandon individual field edits. This feature has now been removed.

4. Fields not requiring amendment may be skipped so as to position at either a later or earlier field in the current record. This feature is also used in order to cycle back round to re-edit earlier fields in the record.

Make a backup

When you have typed in the procedures given in the listings using the Archive 'edit' command, save them with the names given in the initial REMarks. You are advised most strongly to make a backup of these procedures before proceeding. Simply re-save them (or copy them, using the standard QDOS copy command) to a backup microdrive cartridge or floppy disk. It is always good practice with data and program files to make regular backup copies especially where a file is undergoing a large number of changes. This is no less true of data files that are being edited using ArchEd. So backup your dbf files between ArchEd sessions and if possible keep the sessions short. If you have backed up the procedures you are now in a position to continue with ArchEd.

The procedures are arranged in two groups. The first, prefixed by the letters w1, are given in this article. These simply provide you with the mechanism to link your dbf file to the editing procedures. The editing procedures themselves are prefixed by w2 and will appear in the next article.

The main task of the w1 group is to enable individual records selected from your file to be presented to the w2 editing procedures. Most Archive delvers will be acquainted with the techniques employed in the w2 procedures. You may be faced with one of three situations as follows:

1. Firstly, you may wish to use ArchEd to create a new file from scratch. For instance, you may wish to create a diary file where each record represents a 'week' and contains fields 'Mondays\$', 'Tuesday\$', etc. to contain events relating to that week. In this case it is necessary to amend the two procedures prefixed w1b3 in the normal way using the Archive 'edit' command.

```

append
order refno;a
w1z1yn;"Do you want to start entering data? "
while y
w1b3inrec
ink 4
w1z1yn;"Another record? "
endwhile
cls : print "RECORDS CREATED: ";file$;"_dbf = ";count()-1
close
endproc
proc w1b3inrec
cls : use lfn$: last
let refno=refno+1
let field1$=""
let field2$=""
let field3$=""
let field4$=""
append : alter : cls
endproc
proc w1b4opens
print "Opening file..."
open file$ logical lfn$
endproc
proc w1z1yn;p$
local q$
while 1
print p$:: let q$=lower(getkey())
let y=(q$="y")
if instr("ny",q$): print " "+q$: return : endif
print : endwhile
endproc
proc w1z2head
print at 0,0: paper 2:" A R C H I V E   E D I T O R ":
print rept(" ",mod*3); "Memory ";memory(); " Bytes"
if memory()<1200
print at 1,mod*3; "***WARNING** Memory low - Close & backup file!"
endif
endproc

```

```

proc start
REM Save this listing as 'ArchEd2_prg'.
w1a1control
endproc
proc w1a1control
local go0
ink 4
let go0=1
while go0
cls
w1z2head
print at 1,0;"CURRENT RECORD"
w1c2pt
print
w1z1yn;"Do you want to select another record?"
if y
print "Use "; ink 2;"f/n/b/l/q";
print " (first/next/back/last/quit) to select a record."
w1c1vu
print : print : print
endif
print
w1z1yn;"Do you want to EDIT this record?"
if y
run "ArchEd3": cls : ink 4
endif
print
w1z1yn;"Repeat choices?"
if not y: let go0=0: endif
endwhile
print
print "Closing file..."
close
mode 1,mod
endproc
proc w1c1vu
local key$

```

First decide on the content of your file and give suitable names to its field. Write these names down in the order in which they appear in your dbf file record. These names, in this order, then replace the names in the two w1b3 procedures.

Your new file can contain numeric fields but, as mentioned earlier, ArchEd will just ignore these. By running ArchEd's 'w1' procedures you will then be able to create as many initial records as you require by answering ArchEd's questions accordingly. The w2g1 procedure must also be amended before you will be able to insert and edit data in those records (to the full 255 characters available per field) using ArchEd – but more about this in the next instalment.

Acts as a guide

2. If you have an existing file which you wish to edit but it has no supporting application procedures, then the w1 procedures simply act to guide you through to the correct record. This may also be the case if your file has procedures but these are either inappropriate or difficult to amend in order to facilitate ArchEd. In this case it should be quite easy to identify a suitable place in those procedures at which to include a 'run ArchEd'. All that is required then, once file editing is complete, is to return from the ArchEd 'start' procedure, again by the use of a 'run', to the calling procedures. The two w1b3 procedures are unnecessary in these situations and so can be deleted to save space. However, the w2g1 procedure still needs to be amended before proceeding – see next instalment.

3. If you have already developed a set of application procedures to manipulate the records of your file then it should be quite easy for you to dispense with the w1 procedures altogether and link the w2 procedures directly into your existing scheme. If this is the case then tailor your procedures to call the w2a1ed procedure (see next issue) when the record to be edited is current. As in the above cases the w2g1 procedure must be amended.

The following descriptions can be skipped if you're not interested in the inner workings of the file handling procedures.

The individual procedures have the following structure and function:

1. 'a 1 control', as the name suggests, controls the other procedures. It firstly enables the mode to be set (6 or 8, ie tv or monitor). Some parameters used by the w2 editing procedures (col, wth and fl) are then set according to the mode selected. The file to be used is then established by a call to 'blesf'. A 'while' loop then controls the presentation of each selected record to the w2 editing procedures. Records are selected by procedure c1vu. Finally, upon exit from the loop, the file is closed and the standard Archive mode is re-established.

2. 'blesf' controls the establishment of your file. It first calls b2fnm to establish the

```
let key$="z"
while key$<>"q"
let key$=lower(getkey())
if key$="f": first : endif
if key$="l": last : endif
if key$="n": next : endif
if key$="b": back : endif
if key$<>"q":w1c2pt: endif
endwhile
endproc
proc w1c2pt
local i,l,m,n,fnm$,s$
print at 1,0
let n=0
ink 4
while n<3
let fnm$=fieldn(n)
let m=col-len(fnm$)
if fieldt(n)
let s$=fieldv(n)
let l=len(s$)
if l>fl-2: let l=fl-2: endif
let i=fl-l-1
if l<>0
print fnm$:rept(" ",m);": "; ink 6;s$(1 to l);rept(" ",l-1)
else
print fnm$:rept(" ",m);": "; ink 6;rept(" ",1)
endif
else
let i=2
print fnm$:rept(" ",m);": "; ink 6;fieldv(n);rept(" ",1)
endif
let n=n+1
endwhile
endproc
proc w1z1yn:p$
local q$
while 1
print p$:: let q$=lower(getkey())
let v=(q$="y")
if instr("ny",q$): print " "+q$: return : endif
print : endwhile
endproc
proc w1z2head
print at 0,0: paper 2:" A R C H I V E   E D I T O R ";
print rept(" ",mod*3):"Memory ":memory(); " Bytes"
if memory()<1200
print at 1,mod*3:"***WARNING** Memory low - Close & backup file!"
endif
endproc
```

filename and lfn (file\$ & lfn\$). If the file does not exist then b3crf and b3inrec is called to create it. The file is then opened by a call to b4opens.

3. 'b2fnm' is self-explanatory but note that, for ease, it can be altered to reference your favourite file by default.

4. 'b3crf' and 'b3inrec' have been mentioned previously and simply create the file and allow initial records to be inserted using standard Archive commands, eg append, order and alter.

5. 'c1vu' is well-known to most Archivers and allows single key selection of records from the file. 'c2pt' is called to print initial field details on the screen to aid the selection process.

6. 'c2pt' is a universal procedures in that it doesn't rely on the display command or direct use of the file's field names to print the initial contents of the first few fields to the screen. It can be easily altered to display more than just the first three fields if

this isn't enough to uniquely identify your records.

Next month

The file handling procedures themselves allow you to examine your dbf file(s) and/or to create and prepare initial dbf files in readiness for the editor procedures proper which will appear next month. You run the procedures in the normal way by typing 'run ArchEd ENTER'. The questions asked by the file handling procedures are quite straight-forward and need no explanation. Currently, if you answer yes to the question 'Do you want to EDIT this record?' an error will occur when an attempt to 'run ArchEd3' is attempted – so be warned! If this happens, simply issue a 'close' command so as to finish tidily.

Next month we will cover the ArchEditor procedure.

SOFTWARE FILE

INFORMATION

Program: WinBack V1.06

Price: £25 (plus post & packing overseas).

Supplier: Dilwyn Jones Computing, 41, Bro Emrys, Tal-y-Bont, Bangor, Gwynedd LL57 3YT. Tel. (0248) 354023.

WINBACK

A device such as hard disk normally spawns a variety of software. Unfortunately, that hasn't happened with hard disk for the QL. Experience with this device within the QL community is relatively limited, and the anticipated market is small, so programmers left the device alone. There used to be one exception – *Hardback and Finder* – but this pair of programs was available only from PDQL, which was well on the road from the QL scene at the time hard disk drives started to be introduced. Not long afterwards, all but one of the would-be suppliers of hard disk also departed the scene (leastways, they weren't supplying hard disk any more). That left us with Miracle Systems as the sole supplier of a hard disk sub-system, and the only software available was that which was supplied with the drive.

This software was (and is) far from comprehensive. One obvious missing item was a backup routine, to allow the user to make copies of the many hard disk files onto floppies. It is normal with other computers, as well, for such software to be developed by the after-market suppliers, rather than the hardware producers. DJC has now filled the gap by offering *WinBack*. It is planned that an additional program – *WinFind* – will offer the desirable feature, of being able to search for a particular file, or one containing a particular text string.

Backup programs vary in their sophistication, and

Bryan Davies finds a degree of assistance with his hard disk backups.

WinBack is in its early stages yet. One scheduled development is to permit files to be split, so that floppies can be filled and not left with unused space because the next file to be backed up is too big to fit the current floppy. The lack of this feature won't bother users too much, unless they have many – and large – files.

The program requires *Toolkit 2* and the *WIN_REXT* hard disk extensions file to be installed in the QL. The latter is provided with the *Miracle* hard disk and will normally be loaded by its boot routine. *Toolkit 2* is present on hardware such as the *Trump Card*, or it can be purchased separately, on disk or eeprom. (A proposed development of the program is the removal of the need for *TK2*.) The commands used from these two sets of extensions are listed in the instructions (see below). A further set of extensions is provided with the program; the contents of this file will be familiar to many users. Acknowledgements are made to Digital Precision, for the *Turbo* compiler and its toolkit, and to Simon Goodwin's *DIY Toolkit*, as published in *QL World*.

In theory, it should be possible to use the program with hard disk sub-systems other than the *Miracle* one, provided the appropriate extensions are available, but the writer says

that he has not been able to test the program with any other brand of drive sub-system. The required commands that are listed are:

CDEC\$, DATA_USE, FLEN, FOPEN, FOP_DIR, FOP_IN, FTEST, PRINT_USING, PROG_USE, COPY_O, FBKDT, FUPDT, FVERS, SET_FBKDT, SET_FVERS and WIN2.

Note that it might be necessary to use the *WIN2* command after running the program, to get back to where you were before starting, as the program uses this command.

A well-printed instruction booklet is supplied with the program disk. A clone routine is on the disk; the *WCOPY* command can be used as an alternative.

When using the latter, make sure to enter the device names, rather than press ENTER in the hope the ones displayed as examples will be the accepted defaults; simply pressing ENTER may not achieve the desired results. If the program is run from a floppy disk, the *WIN_REXT* file needs to be copied to that disk from the hard disk.

The short boot file could be integrated into your normal hard disk boot, and the usual *DEVICE\$* statement altered, if *WinBack* is to be run from hard disk. A routine is provided for the user to set suitable defaults,

if necessary, and defaults can be changed at run time also. You are advised to run the program from floppy, although it is not made clear why ('To avoid any memory constraints' is mentioned).

The F1 selection from the 'Sinclair' screen should be made when starting the boot; the program windows are suitable for both monitor and tv. The main menu is simple – there is a choice of the credits for the contributors to the writing of the program, setting the QL clock, starting the backup, or quitting the program. Rather unusually, the writer of the program supplies his name and address and invites letters containing suggestions or complaints; it is more usual for programmers to remain relatively anonymous, shielded by the supplier. Presumably, there won't be a shoal of letters, as there are not all that many hard disks running on QL systems. It is pointed out in the instructions that the setting of the clock is very important in the context of backing-up files; the obvious point is that you back up files by their date stamps, and it's not much use trying to do that if all the files are stamped '1-1-61', or something similar.

The backup sub-menu has six options, selected by the function keys and the ESC key. They are:

F1 SET OUTPUT DEVICE
F2 SET PRINT DEVICE
F3 SET TOP DIRECTORY
F4 PROCEED WITH BACKUP
F5 VIEW SYSTEM DEFAULTS
"ESC RETURN TO MAIN

MENU"

The F1 option allows the user to specify where backup files are to be placed. You are not constrained to put them on floppy disk, and alternatives are hard disk, ram disk, microdrive or some as-yet-unknown device. Only five characters are permitted for the name of the output device, and they must include the underscore, so you can't use a sub-directory name. With F2, you can choose to print to a printer or to a file, but a file specified should not be in a hard disk directory which is going to be backed up.

The option is offered to have all the backed-up files listed, together with the device to which they are sent; if this is not chosen, nothing is printed. The baud rate setting for transfer of data to the printer can be any of the usual values, up to 19,200. The F3 option gets into the important area for the hard disk files – sub-directories. The 'top' level has to be specified, the default being the Root Directory, WIN1_.

If a particular sub-directory, and any other directory levels below it, are to be backed-up, the name of that sub-directory is entered, eg WIN1_QUILL_. The underscores matter; there must be one in the fifth charac-

ter position, and at the end of the string. In addition, the fourth character must be in the range 1-8. The next option allows you to restrict the sub-directory range to just the top level already specified, or to include all sub-directory levels below that as well. F5 will display a list of the set defaults.

This brings you to the actual work. F4 starts the backup process. Elementary perhaps, but it's nice to have both a visual and an audible reminder that the first disk of your backup set should be put into the appropriate drive, to enable the backup to start.

You are instructed to number the disks in sequence 1,2,3, etc. As each file is backed-up, the details are displayed on the screen, and on the printout (if that has been requested). File name, size, date of last modification, and date when last archived are displayed, followed by the number of files so far processed, their corresponding byte-total, and the space still available on the current backup disk. The directory name, and level, and the number of directories processed so far, are listed. The screen display is well laid-out, and clear. The printout starts with the date and lists each backed-up file, together with the name of the

directory it came from and the device it has been backed up to.

When the current backup disk has been filled, a warning message (see illustration) is displayed, and an audible warning is given too. The next disk is inserted and ENTER pressed, and the process continues. It took about six minutes to fill one disk completely – 0 bytes free – so you can't go far away if you want the backup to be finished as soon as possible; however, your presence is not required other than to feed in disks and you carry on reading the paper, writing your latest masterpiece on the 'other' QL, or making a quick cup of coffee. Very roughly, you need to allow yourself about 5-6 hours to back up a full 40 MB hard disk, *the first time*. As subsequent backups will only process those files which have been updated since the original backup, the time taken will normally be very much less. That is, one good long initial session sets the basis for future, *regular*, short sessions.

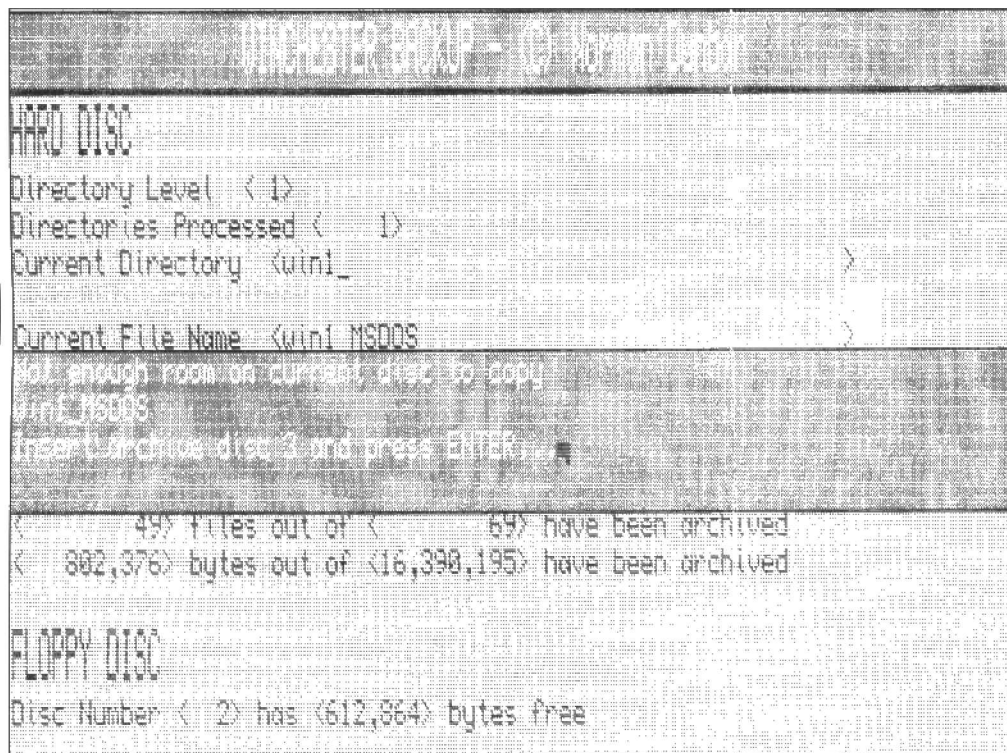
The time taken to fill a backup disk is very dependent on the size of files being backed up. For example, if the files the program encounters happen to be a bunch of small ones followed by one very big one, it may be that the program will

tell you another disk is required within a minute or so, because there isn't enough room on the disk for the big one. The program, in its current form, cannot split files between backup disks. This may not matter to most users, but these who habitually create big files will find they need (relatively) many more disks. The program writer has indicated that he intends to add the facility to split files at a later stage in the development. My experience with backing-up files on a PC suggests that file-splitting is very important to 'heavy' computer users.

Another, related function that is not currently incorporated is file compression. As it stands, the program makes a size-for-size copy of files, so that 40 MB of hard disk files require at least that amount of floppy disk space for backup. That is, a full Miracle hard disk needs more than 50 (720 KB) floppies for a complete backup. At current prices, the cost of this many disks is not too bad, and they will fit into one disk box, but some users will no doubt feel that compression is desirable. Typically, a decent compression routine – in conjunction with the ability to split files – can reduce the space required by 30-50%, with a corresponding reduction in the time spent. The snag with it is that all the files are normally concatenated; that is, as many files as possible are combined and put into one big file per disk. This means you cannot locate the backup of one particular file in a hurry if you need to to replace the current one on the hard disk.

At the current stage of development of QL add-ons, it seems quite acceptable not to have the more advanced features found in some other computer software, and lack of file-splitting and compression in WinBack should not worry QL hard disks users unduly.

It is desirable to plan your backup in advance. You can't step out of WinBack (it is started by EXEC_W) and format a few more disks if you find yourself short of enough empty ones to complete a backup. A restriction that appeared before I had considered the possibility of it was that created by partitioning the hard disk. In my case, 15 MB of the disk were allo-



cated to *PC Conqueror*, to run MS-DOS.

As far as the backup program could see, this whole area of storage was just one file, and it asked for another disk to put it on. Not surprisingly, I didn't have a 15 MB floppy to hand (disks of around this size are said to be available for about £30, but not from my local shop!), which meant that the backup had to be aborted by resetting the QL, but that didn't affect the files already backed up.

This problem can't be side-stepped by specifying the MS-DOS partition as a sub-directory that should not be backed up, since the whole partition is treated as just one file, not a sub-directory. If all "normal" files come before it on the disk, there will not be a problem backing them up, but any that come after it can't be processed. The effect is basically the same if a normal QDOS file is larger than 717 KB, which is possible when you use hard disk as the 'save' medium for a word-processing program, for example.

Many users will not have

had occasion to read-up on the details of archiving and date-stamping files. Fortunately, the instructions deal with some of the important points. The screen which shows the progress of a backup displays FILE LAST UPDATED and FILE LAST ARCHIVED. The former is followed by the date on which the current file was last modified, and the date is taken from the system clock at the time of modification of the file; if you failed to set the system clock correctly, the date will be incorrect. *WinBack* is not to know whether or not you bother to set the clock each time you start the QL. It has to assume the date it finds is correct.

The ARCHIVED date is the date on which a file was last backed by the *WinBack*. This date also will be wrong if you failed to set the clock before doing the backup. It is quite possible that the updated date will show as later than the archive date when the reverse is true, if you have failed to set the clock at the appropriate times. You could find all files being backed up every time, even though none actually need it. **Make sure you**

set the QL clock - every time!

In general, users are likely to be satisfied with the program's decision to back up only those files which have been updated since the last backup. Hopefully, future developments of the program will give the user more freedom to choose, such as allowing *any* files to be specified for backup, regardless of when they were last backed up. It is also desirable to be able to specify groups of files; as it stands, program files are backed up as well as data ones and the user may wish to keep these two categories separate from each other. (You may keep them in separate sub-directories, however, in which case there should be no problem.)

When the time comes to copy a file back to the hard disk, you may appreciate having made a printout during the backup, because it can be rather a pain to have to insert disk after disk and do DIRs simply to find out which disk an odd file is on. As the files are not compressed, or otherwise tampered with, the COPY or WCOPY commands will retrieve the required file(s) with-

out bother.

If I seem not to have said much about this program, that should be taken as a compliment to it. It does what it sets out to do, and the user does not have to be particularly smart to use it. The price is reasonable. The instructions are clearly-printed (no bother with printing-out a DOC file, other than any UPDATES.DOC) and user-friendly. The type and manner of the backup will be quite satisfactory for most users. Future developments should cater better for the more-demanding user, but the program in its present form should, at the least, ensure that the big disasters (often) associated with hard disk usage are avoided. As mentioned in the instructions, it is not a question of *whether* you will have file problems - just *when*. Most times, you will have created the problem yourself (which is no consolation). It is most reassuring to know you have a spare, up-to-date, set of files ready for copying back onto the hard disk.

C.G.H. SERVICES

Cwm Gwen Hall, Pencader, Dyfed, Cymru, SA39 9HA (Tel. 0559 384574) (1pm - 9pm)

COMMERCIAL SOFTWARE

ANELPUM QUAT (NICK WARD) (text adventure) (128k) (flp/mdv) £10.00
 ASSAULT AND BATTERY (DAMON CHAPLIN) (shoot 'em up) (128k) (flp/mdv) £12.50
 ASTRO (NICK WARD) (astrology prog) (128k) (flp/mdv) £12.50
 THE BLAG2 (TONY WOOLCOCK) (text adventure) (256k) (flp/mdv) £10.00
 D-DAY MKII (DDC & RICH MELLOR) (wargame) (256k) (flp) £15.00
 DOUBLE BLOCK (FRANCOIS LANCAULT) (arcade game) (128k) (flp/mdv) £10.00
 DREAMLANDS (JEAN-YVES ROUFFIAC) (text adventure) (256k) (flp) £10.00
 FIVE GAMES PACK 1 (WREIFORD DAVIES) (mind games) (128k) (flp) £12.50
 FROM THE TOWER OF VALAGON (ALAN PEMBERTON) (text adventure) (128k) (flp/mdv) £10.00
 GEE-GEE SYSTEM (PHIL JONES AND ANDY CSERBAKOL) (race predictor) (128k) (flp/mdv) £10.00
 GREY WOLF (OLIVER NEEF) (u-boat simulator) (256k) (flp) £10.00
 HERE WE GO (PHIL JONES AND ANDY CSERBAKOL) (text adventure) (128k) (flp/mdv) £10.00
 MACSPORRAN'S LAMENT (DAVE WATSON) (illustrated text adventure) (128k) (flp/mdv) £10.00
 OPEN GOLF (OLIVER NEEF) (golf simulator) (384k) (flp) £12.50
 ORBITING STARS (JOHN TOPHAM) (astronomical simulator) (128/256k) (flp/mdv) £10.00
 PERSONAL FINANCE MANAGER (JASON VICINANZA) (budget system) (128k) (flp/mdv) £10.00
 POLYTEXT (NICK WARD) (multi-column quill) (128/256k) (flp/mdv) £17.50
 PUDDGE (DAMON CHAPLIN) (arcade game) (128k) (flp/mdv) £12.50
 QUICK MANDELBROT (KENNETH MURRAY) (fractals) (128k) (flp/mdv) £10.00
 QUICK MANDELBROT & JULIA (KENNETH MURRAY) (fractals) (128k) (flp/mdv) £12.50
 QUIZMASTER (PHIL JONES AND ANDY CSERBAKOL) (quiz game) (128k) (flp/mdv) £10.00
 QUIZMASTER II (AS ABOVE & RICH MELLOR) (quiz game + module option) (128k) (flp/mdv) £12.50
 RETURN TO EDEN (OLIVER NEEF) (role-playing illustrated adventure) (256k) (flp) £17.50
 SECTOR X (HORST SPIERLING) (shoot 'em up) (256k) (flp) £12.50 (mdv) £15.00
 SPEEDFREAKS (DAMON CHAPLIN) (car race game) (128k) (flp/mdv) £12.50
 SQUIDGY ROUND THE WORLD (MICHAEL CROWE) (arcade game) (flp) £12.50 (mdv) £15.00
 STARPLOD (ALAN PEMBERTON) (icon-driven adventure) (128k) (flp/mdv) £10.00
 UNCLE LOONIE'S LEGACY (DAVE WATSON) (puzzle adventure) (128k) (flp/mdv) £10.00
 VOYAGE OF THE BEANO (ALAN PEMBERTON) (illustrated text adventure) (256k) (flp) £12.50
 WRECK DIVE (NICK WARD) (arcade adventure) (128k) (flp/mdv) £10.00
 CLIP ART 1 - SPORTS (FLP) £6.00
 CLIP ART 2 - WHIMSIES (FLP) £6.00
 CLIP ART 3 - OFFICE (FLP) £6.00
 CLIP ART 4 - VIZ (FLP) £6.00
 CLIP ART 5 - GENERAL (FLP) £6.00

SPEEDSCREEN - ROM VERSION £20.00
 SPEEDSCREEN - FLP/MDV £10.00

PUBLICATIONS

QL TECHNICAL REVIEW ISSUES 1 - 2 £1.50 EACH
 QL TECHNICAL REVIEW ISSUES 3 - 6 £1.75 EACH
 QL ADVENTURERS FORUM ISSUES 1 - 3 £1.25 EACH
 QL ADVENTURERS FORUM ISSUES 4 - 9 £1.50 EACH
 LEISURE REVIEW ISSUE 1 £1.75 EACH
 INTERNATIONAL QL REPORT ISSUES 1 - 2 £1.50 EACH
 QL SURVIVOR'S SOURCE BOOK £2.50 EACH

ALL PRICES INCLUDE MEDIA AND POST AND PACKING FOR THE UK.
 PLEASE ADD 10% FOR ORDERS IN EUROPE; 20% OUTSIDE OF EUROPE.
 PLEASE INDICATE 3.5"/5.25"/MDV VERSIONS WHEN ORDERING

PUBLIC DOMAIN AND SHAREWARE LIBRARY ALL DISKS £2.00 EACH INCLUSIVE OF MEDIA AND P&P

ADVENTURE GAMES DISK 1 (includes fantasia and ye classical type adventure)
 ADVENTURE SOLUTION DISKS 1-3 (includes The Pawn and Mortville Manor, mainly ST though!)
 ADVENTURE SOLUTIONS DISK 4 (QL specific)
 ADVENTURE GAMES SOURCE CODE DISK 1 (includes Fantasia, Haunted House, etc.)
 ADVENTURE UTILITIES DISK 1 (includes Quill to SuperBasic converter and demo adventure)
 AUSTRALIAN P.D. DISKS 1 - 2 (include a wide variety of games, utilities etc.)
 COMMUNICATIONS DISK 1 (includes QBOX Bulletin Board system)
 COMMUNICATIONS DISK 2 (includes QL Kermit)
 CONNECTIONS DISK 1 (includes ST, QL screen and file converters)
 DEVICE UTILITIES DISK 1 (includes Archiving, compacting, formatting and Shell programs)
 DILWYN JONES DISK 1 (includes many SuperBasic progs and Wordsearch)
 EDITORS DISK 1 (includes QED and MicroEmacs)
 EDUCATIONAL PROGRAMS DISK 1 (includes maths, music, and chemistry programs)
 EMMANUEL VERBECK DISK 1 (includes screen save and print progs, and many more utilities)
 ESOTERICA DISK 1 (includes Div Pyramid construction prog, Biorhythms and Psychology progs)
 FRACTALS DISK 1 (includes large numbers of mandelbrot and other fractal progs)
 FRACTALS DISK 2 (Carl Cronin's Mandelbrot prog plus animation screens)
 FRACTALS DISK 3 (Rainer Kowalik's mandelbrot prog as amended to give 'Jewels' effect)
 GAMES DISK 1 (includes Starburst, Cavern, Frenzy)
 GAMES DISK 2 (includes many arcade type games)
 GAMES DISK 3 (includes QL War)
 GRAPHIX ANIMATION DISK 1 ('Movieola')
 GRAPHIX DEMOS DISK 1 (too many to list!)
 GRAPHIX SCREENS (GIF FORMAT) DISKS 1, 2, 3
 GRAPHIX SCREENS (PIX FORMAT) DISK 1
 GRAPHIX SCREENS (SPECTRUM) DISK 1
 GRAPHIX SCREENS (ST) DISKS 1, 2, 3
 GRAPHICS-SCREEN UTILITIES DISK 1 (includes sprite designer, CAD and windows progs.)
 HAM RADIO DISK 1 (includes PC conversions)
 INDEXES DISK 1 (includes QL World and QUANTA indexes)
 MATHS AND CALENDAR DISK 1 (includes calculator, pi and calendar creator progs)
 OLIVER FINK DISK 1 (includes progs requiring Pointer environment to work)
 PRINTER UTILITIES DISK 1 (includes label creators and printer tutorials)
 PROGRAMMING DISK 1 (includes QL PROLOG)
 PROGRAMMING DISKS 2 - 8 (C88 Compiler)
 RALF BIEDERMANN DISKS 1 - 2 (excellent collection of programs, games, utilities etc.)
 RECREATIONAL MATHS DISK 1 (includes life cellular automata etc.)
 SUPERBASIC UTILITIES DISK 1 (includes large numbers of utility progs, toolkits, procedures etc.)
 TEXT DISKS 1 - 7 (The Bible)
 TEXT DISKS 8 - 9 (Computer Jargon)
 TEXT DISK 14 (400+ Business Letters from U.S.A.)
 Please note the above are only part of our P.D. Library. New programs are constantly being added and more are always welcome. Please send an S.A.E. for full catalogue. Please add postage as for commercial software.
 We can also supply commercial software for STs and Amigas!

SUBSCRIBE TO *Sinclair QL World* AND GET THIS GREAT BINDER...



...ABSOLUTELY FREE!!

Fill in the coupon below and send it with your remittance to: MSM
Subscriptions Department, Lazahold Ltd, P.O. Box 10, Roper Street, Pallion
Industrial Estate, Sunderland SR4 6SN. Telephone 091 510 2290.

(The first issue of a new subscription to be delivered will be one or two issues after the one you placed your order in.)

Please start/renew my 12/24 month subscription to SINCLAIR QL WORLD and send my free binder.		
I enclose my cheque/money order for £ _____ made payable to MCPC Ltd or debit my Access/Visa card.		
No _____	Expiry date _____	
Name _____	Address _____	
_____ (Please enter postcode to ensure prompt delivery)		
Signed _____	Date _____	SQL 0991

RATES

UK £23.40 12 mth
£46.80 24 mth

Europe £32.90 12 mth
£65.80 24 mth

Rest of £40.90 12 mth
World £81.80 24 mth

OVERSEAS RATES
INCLUDE AIRMAIL SERVICE